



Universidad
Carlos III de Madrid
www.uc3m.es

TRABAJO FIN DE GRADO

**Estudio aplicabilidad SDN en red de agregación
EoMPLS**

Memoria

Autor: Jorge Prudencio Fernández

Titulación: Grado en Ingeniería Telemática

Tutor: Jose Alberto Díaz García

Director: Carlos Jesús Bernardos Cano

Fecha: 24 de Septiembre de 2014

Índice de contenidos

1. Resumen

2. Introducción

3. Motivación y objetivos

3.1. Motivación.....	7
3.1.1. Internet: Un servicio en crecimiento continuo	7
3.1.2. Requisitos del tráfico multimedia	8
3.1.3. Las demandas de los usuarios	9
3.1.4. Las redes actuales.....	10
3.1.5. ¿Se necesita un cambio?	11
3.2. Objetivos.....	12

4. Planteamiento del problema: análisis del estado del arte, requisitos, restricciones y marco regulador

4.1. El problema.....	13
4.2. Análisis del estado del arte.....	15
4.2.1. Las redes actuales: basadas en MPLS.....	15
4.2.2. VPNs, la necesidad de interconectar redes separadas entre sí	16
4.2.3. EoMPLS: Ethernet sobre MPLS.....	16
4.2.4. La alternativa: SDN	18
4.2.5. Los beneficios de SDN [13] [14].....	19
4.3. Requisitos	20
4.4. Restricciones.....	20
4.4.1. Restricciones generales para los operadores.....	20
4.4.2. Los fabricantes de hardware de red.....	21
4.5. Marco regulador.....	21

5. Diseño de la solución técnica

5.1. Objetivos.....	23
5.2. OpenFlow: La implementación de SDN.....	23
5.3. ¿Cómo funciona OpenFlow? [17] [18] [19]	23
5.4. Elementos utilizados.....	28

5.4.1. Bloque 1	28
5.4.2. Bloque 2	28
5.5. Estructuras de red	29
5.6. Características del controlador.....	31
5.6.1. Detección de estado de enlaces.....	31
5.6.2. Merging.....	32
5.6.3. Simplicidad en inserción de flujos	32
5.6.4. Estructura del código y reutilización del controlador	33
5.7. Casos de testing.....	34
5.7.1. Caso 1: Red simple con 2 Switchs y 2 Hosts (Mininet) + Controlador Pox	34
5.7.2. Caso 2: Red media con 4 Switchs y 2 Hosts (Mininet) + Controlador Pox (camino de backup)	37
5.7.3. Caso 3: Red grande con 6 Switchs y 3 Hosts (Mininet) + Controlador Pox (camino de backup y merging)	39
5.8. Descarga de controlador	43
5.9. Configuración de router Linksys WRT54GL [20] [21]	43

6. Resultados y evaluación

6.1. Ventajas detectadas	46
6.1.1. Tiempo de adaptación	46
6.1.2. Visión general de la red	47
6.1.3. Redes de reducido presupuesto.....	47
6.2. Desventajas detectadas.....	47
6.2.1. Mayor sensibilidad frente a ataques.....	48
6.2.2. Protocolo muy joven	48
6.2.3. Necesidad de análisis de paquetes.....	48
6.3. SDN: ¿Un candidato para el cambio?	49

7. Presupuesto y planificación del trabajo

7.1. Presupuesto.....	51
7.2. Planificación.....	52
Estructura de trabajo.....	52
Fases y sub-fases	53

8. Conclusiones

8.1. El futuro de SDN	57
8.2. Aprendizaje adquirido	58
8.3. Próximos pasos	58
Desarrollo de API REST	58
Gestión de encendido/apagado de nodos	59
Creación de VPNs diferenciadas.....	60

9. Bibliografía

10. Anexos

1. Resumen

La forma de trabajar en Internet, tanto de usuarios como de empresas, ha cambiado en los últimos años. Se está pasando de un modelo en el que basado en proveedores de datos que residían en partes localizadas de la red a otro en el que cualquier punto, en cualquier momento puede ser el origen de grandes cantidades de datos.

Este cambio de modelo está obligando a que los operadores comiencen a plantear soluciones escalables que soporten las tendencias futuras de tráfico en Internet. Se prevé que dentro de cuatro años [1] el movimiento de datos en Internet se triplicará, y debemos estar preparados para ello.

Las empresas proveedoras de Internet están buscando un modelo lo suficientemente flexible que permita por un lado, asegurar sus negocios, y por otra soportar cambios rápidos en las redes actuales y dar cabida a las necesidades de los usuarios de forma rápida y eficiente.

Actualmente las tecnologías disponibles están dejando de ser poco a poco efectivas ya que son sistemas bastante rígidos y que no tienen el suficiente dinamismo para soportar las exigencias de los clientes, sobre todo las futuras.

Aquí es donde entra en escena SDN, un nuevo paradigma que cambia completamente la forma de entender las redes. Hasta ahora el funcionamiento de las redes de acceso a Internet funcionan de forma distribuida. La infraestructura está formada por Switchs que se comunican entre sí y que controlan de forma conjunta el tráfico de Internet.

Sin embargo, este modelo esté dejando de ser factible, ya que cada Switch debe ser configurado de forma individual para afrontar cambios en la red.

SDN propone un modelo centralizado en el que se separa el plano de control, basado en un controlador que reside en un punto concreto de la red, y el plano de datos, basado en la infraestructura de Switchs.

Este cambio de filosofía viene a solucionar el problema de flexibilidad existente actualmente ya que permite tener una visión global de la red desde un punto y gestionar la red desde él a través de algoritmos más inteligentes basados en software.

En este proyecto se tiene como objetivo revisar las ventajas teóricas a través de casos prácticos, basados en redes reales de operador, y así ser capaces de obtener una idea más exacta de cuál puede ser el futuro de esta tecnología y comprender mejor las ventajas que ofrece.

2. Introducción

Hoy en día nadie duda que la tecnología e Internet son los dos elementos en los que se basa nuestra vida actualmente y lo estará aún más en el futuro. Comunicación, información, entretenimiento, seguridad, banca, orientación, compra, estadística, multimedia,... todos estos ámbitos y un incalculable número más han sido trasladados a Internet y ya no se concibe el mundo sin él.

Tal es la importancia que ha adquirido Internet que es esencial asegurar un servicio continuo y de calidad tanto a clientes particulares como empresas. Internet es una herramienta en la que se basan cada vez más negocios y una caída del servicio puede suponer grandes pérdidas.

Sin embargo, nos encontramos solo al principio de lo que será un futuro basado completamente en Internet. La necesidad de estar conectados e informados todo el día está cada vez más latente en la sociedad.

Para soportar esta gran demanda futura de Internet los operadores han empezado a trabajar e investigar formas que les permita asegurar el acceso a la red y así responder a la necesidad de sus clientes.

Generar esta respuesta no es fácil, ya que se deben utilizar soluciones que permitan generar un modelo de negocio rentable para los operadores y que tengan en cuenta las tendencias de precios y los hábitos de los usuarios de Internet.

Por ello, es importante empezar ya a investigar nuevas tecnologías y filosofías de red que permitan estar preparados para el futuro de Internet. Los operadores se han convertido en empresas muy importantes en nuestros días, siendo los responsables de que todo el mundo tenga acceso a Internet. Ser el encargado de dar acceso a miles de usuarios a un servicio tan básico obliga a que cada cambio deba ser probado estrictamente antes de ponerse en el mercado.

Durante este documento se recogerá la investigación de SDN, la nueva alternativa que tiene como objetivo evolucionar el modelo distribuido tradicional de las redes a un sistema centralizado en el que se obtiene una visión global de la red.

3. Motivación y objetivos

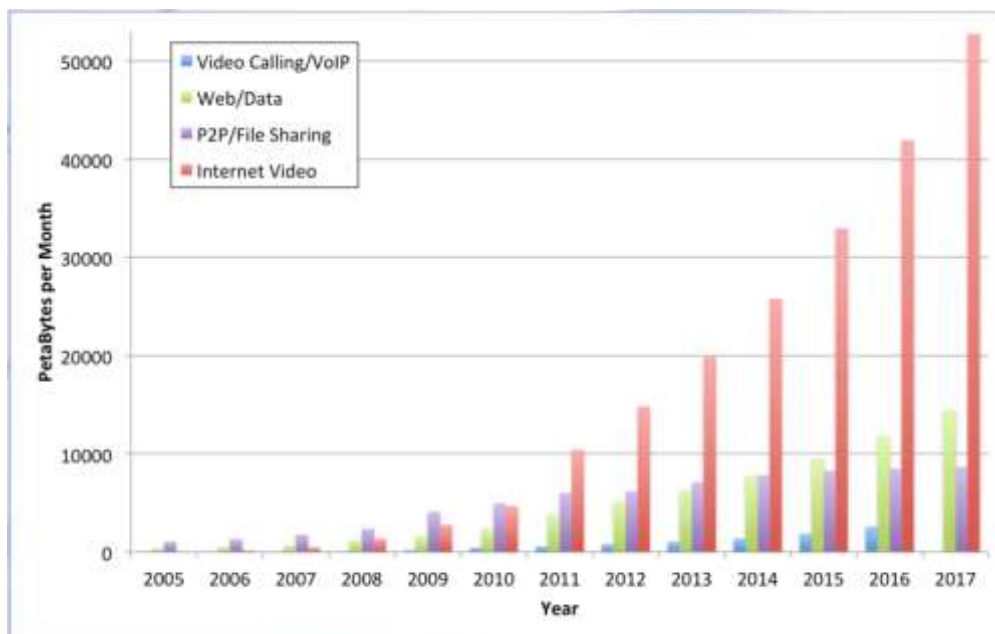
3.1. Motivación

3.1.1. Internet: Un servicio en crecimiento continuo

Las redes de Internet están evolucionando mucho en los últimos años. Las necesidades y costumbre de los usuarios han cambiado y Internet cada vez está más integrado en nuestra vida.

Antaño se realizaba un uso de las redes relativamente pequeño ya que destacaban sobre todo los servicios de mailing, el uso de webs sencillas únicamente programadas en HTML y la descarga de pequeños datos (fotos o documentos).

Esta situación ha ido cambiando y se prevé que cambie aún más debido a la tendencia que lleva Internet de que la mayoría de tráfico sea multimedia. Si en 2012 el tráfico multimedia se situó en 14.818 PetaBytes al mes, triplicando la cifra de tráfico web y email, en 2017 se prevé que el tráfico multimedia en 52.752 PetaBytes al mes, aumentando de forma exponencial año por año.



Consumer Internet Traffic						
	2012	2013	2014	2015	2016	2017
By Subsegment (PB per Month)						
File Sharing	6,201	7,119	7,816	8,266	8,478	8,667
Internet Video	14,818	19,855	25,800	32,962	41,916	52,752
Web, email and data	5,173	6,336	7,781	9,542	11,828	14,494
Online gaming	22	26	32	39	48	59
Voice over IP (*)	154	159	163	169	174	
Total (PB per Month)						
Consumer Internet Traffic	26,213	33,337	41,429	50,809	62,269	75,973

From Cisco Visual Networking Index: Forecast and Methodology

Este aumento exponencial del tráfico multimedia deberá ir acompañado por un cambio de las redes que actualmente utilizamos para soportar la demanda de datos. Un gran aumento de datos en las redes actuales podría provocar que se saturasen enlaces con mucha más frecuencia y los enrutadores no fueran capaces de soportar el gran volumen de datos al que serían sometidos.

Para dar soporte al gran volumen de datos al que serán sometidas las redes en el futuro se podría optar por varias soluciones:

- Invertir en la compra de nuevos equipos que realicen un enrutado de paquetes más rápido y de forma más fiable. Lo que conllevaría una gran inversión por parte de los operadores que posiblemente iría acompañada por una subida de precio de los servicios de Internet.
- Invertir en mejorar la eficiencia de las redes actuales que actualmente no están siendo utilizadas al 100% (sobre todo en momentos de congestión). Esto se podría conseguir a través de la investigación de nuevas técnicas y protocolos de enrutado de datos. Con esta opción haría falta una inversión progresiva de dinero para investigación que no repercutiría tan notablemente en forma de coste por servicio para el usuario final.

3.1.2. Requisitos del tráfico multimedia

Como se ha expuesto antes, una gran cantidad del tráfico que se encontrará en nuestras redes en un futuro será multimedia. Este tipo de paquetes tienen unos requisitos [3] especiales que los diferencia del resto de tráfico:

1. Sensibilidad a jitter: Una variación continua del jitter (variación del tiempo que tarda un paquete en ir del origen al destino) puede resultar un gran problema en tráfico multimedia y crear problemas en el servicio. Si los paquetes multimedia no llegan de forma continuada unos tendrán que esperar a otros provocando desfase entre emisor y receptor. La variación de jitter se agrava sobre todo en condiciones de saturación de la red.
2. Flujos largos y continuos: Al contrario de tráfico como el web que suelen ser intermitentes y en los que se finaliza la conexión rápidamente, el tráfico multimedia puede tener flujos continuos de varios minutos u horas con lo que se hace un uso más intensivo de la red.
3. Varios receptores/emisores simultáneos (multicast): Con la llegada de servicios como las multi conferencias o la emisión en streaming se ha creado la necesidad de enviar paquetes desde varios orígenes a varios destinos a la vez lo que aumenta aún más la demanda de recursos.
4. Exigencia de gran calidad en el servicio: El usuario cada vez demanda una mayor calidad de imagen y de sonido. Servicios como Youtube ya permiten subir video desde 144p a 1080p. Si deseamos realizar una videoconferencia a una resolución de 1080p sería necesaria un ancho de banda mínimo de 2 Mbps en subida y bajada [2]. Si revisamos las conexiones que ofrecen los operadores actualmente deberíamos contratar una conexión de 20Mbps de bajada / 2Mbps de subida para poder soportar una única conferencia.

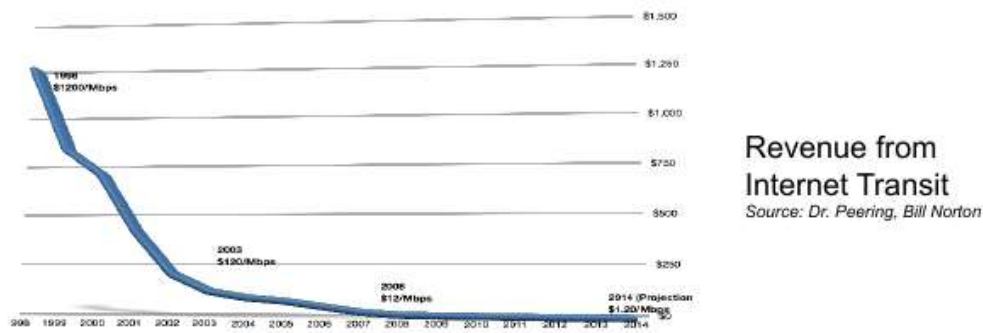
Estos estrictos requisitos que el tráfico multimedia tiene son el principal motivo que lleva a necesitar redes rápidas y eficientes que garanticen un servicio multimedia de calidad y sin retrasos.

3.1.3. Las demandas de los usuarios

Todos los cambios que se realicen en el futuro en las redes se verán fuertemente influenciados por las exigencias de los usuarios en cada momento. El objetivo final de una empresa es generar beneficios a través de sus operaciones y para ello necesita crear productos atractivos a los clientes.

Si echamos un vistazo a la tendencia de precios de internet a lo largo de los años observamos una tendencia clara:

Market Forces in New Internet



Evolución de los precios de Internet en los últimos años [4]

Como se puede observar el precio del Mbps ha disminuido exponencialmente desde la apertura del Internet al público.

Además la demanda de conexión por parte del usuario final también ha cambiado. Si hace 5 años nos bastaba con una conexión de 10-20 Mbps para realizar un uso normal de Internet, actualmente ha crecido la demanda de conexiones de 50-100 Mbps para servicio residencial. Algunos operadores ya han decidido preparar nuevos servicios de 200 Mbps [5] para adelantarse a la futura demanda que existirá en los usuarios dentro de poco.

Tras estos datos, podemos concluir que la tendencia que se está siguiendo en el mundo de las conexiones de red es a ofrecer cada vez un servicio más barato y de mayor calidad.

3.1.4. Las redes actuales

Las redes de Internet han evolucionado mucho en los últimos años permitiéndonos obtener grandes velocidades de acceso y transferencia de datos. Actualmente disponemos redes que permiten ofrecer hasta 200Mbps en hogares y conexiones de hasta 100Gbps para empresas en servicios como MacroLan de Telefónica [6].

Esta gran evolución se debe sobre todo a la incorporación de la fibra óptica por parte de los operadores que permite llegar a velocidades incluso mayores a las disponibles.

Sin embargo, aunque la velocidad en la conexión entre equipos ha aumentado de forma considerable, existen aún limitaciones en los nodos de interconexión.

Estas limitaciones se deben a dos motivos principales:

- Los equipos de enrutamiento funcionan con circuitos electrónicos convencionales con lo que la velocidad de enrutamiento en algunas ocasiones es inferior a la velocidad de llegada de paquetes.
- Los protocolos actuales no aprovechan al 100% los recursos de la red ya que tienen una visión parcial de la red.

Estas limitaciones se manifiestan sobre todo cuando existe una exigencia grande de recursos en la red. Por ejemplo, en momentos en los que se consulta un contenido multimedia de forma masiva por millones de usuarios es muy común que se produzcan retrasos o parones en los contenidos.

3.1.5. ¿Se necesita un cambio?

Si enumeramos las tendencias del mercado actual de los operadores de red, expuestas anteriormente, detectamos por lo tanto:

- El tráfico de Internet crece de forma exponencial. Internet está cada vez en más ámbitos de nuestra vida y el usuario tiende a usar cada vez más aplicaciones y servicios multimedia.
- Los usuarios demandan cada vez conexiones más rápidas y más baratas.
- Los servicios multimedia necesitan unas condiciones muy concretas para funcionar de forma correcta (bajo jitter, mayor ancho de banda,...).
- Los protocolos de enrutamiento actuales no aprovechan completamente los recursos de las redes actuales.

A través del estudio de estas tendencias de mercado y demandas del tráfico que predominará en el futuro (multimedia) nos encontramos con que es necesaria una evolución de las redes actuales para que se realice un mayor aprovechamiento de los recursos que ofrece.

La solución adoptada además debe conservar la tendencia de precios a la que el usuario final está acostumbrado, descartándose soluciones que puedan conllevar una gran inversión inicial y por consiguiente una subida considerable de los precios. Por eso serán preferibles las soluciones de redes con menor Capex y menor Opex.

Además es muy importante que la decisión que se adopte empiece a ponerse en práctica cuanto antes ya que como se ha visto anteriormente el crecimiento de datos en Internet va a ser muy rápido, las redes deben estar preparadas para soportarlo y los operadores necesitan mantener un balance ingresos/gastos que haga sus negocios sostenibles.

3.2. Objetivos

El objetivo principal de este trabajo será investigar y trabajar con una de las alternativas que más están sonando actualmente para gestionar el tráfico de las redes actuales: las redes SDN (Software-Defined Networking).

SDN se ha propuesto como un paradigma que permitiría aumentar la eficiencia de las redes pasando de un modelo de control distribuido a uno centralizado en un punto. Con esto se conseguiría obtener una visión completa de la red y crear algoritmos que aprovechen mejor los recursos [7].

Durante el trabajo se explicará en que consiste SDN y cuál es la idea principal que se sigue en redes basada en él. Posteriormente, se analizarán las ventajas y desventajas teóricas de utilizar SDN.

Una vez comprendida la tecnología se procederá a realizar el montaje práctico de varias redes basadas en el paradigma y que son comparables a las redes utilizadas por los operadores actualmente. Todo esto con el objetivo de validar las ventajas y carencias de forma práctica y contar con datos que nos permitan hacer una previsión del futuro de SDN en el control de las redes de los proveedores de red.

4. Planteamiento del problema: análisis del estado del arte, requisitos, restricciones y marco regulador

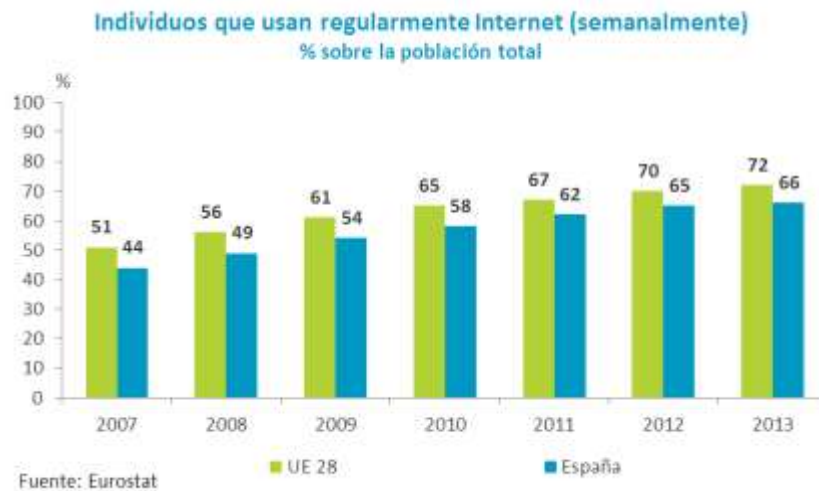
4.1. El problema

Como se exponía antes el inminente crecimiento del tráfico multimedia en las redes actuales está obligando a mejorar la eficiencia de las redes actuales.

En especial este gran crecimiento de volumen de datos en Internet se debe al aumento del tráfico multimedia. Nuestros hábitos y costumbres tienden a hacer un uso muy intensivo de este tipo de tráfico que requiere mucho ancho de banda y un jitter reducido para funcionar correctamente

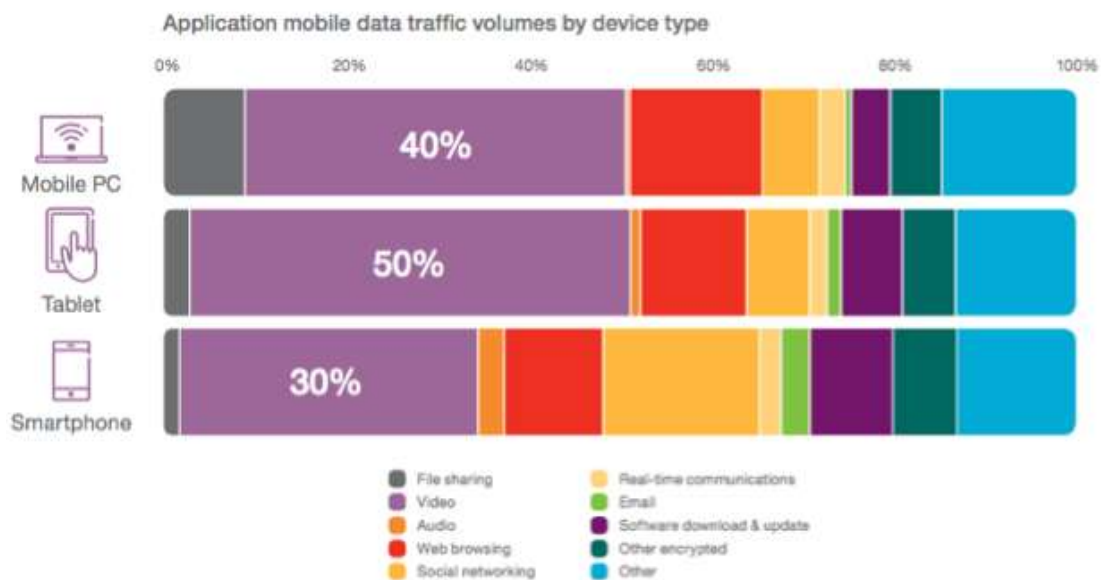
Algunos hábitos que han desencadenado esta tendencia de aumento de datos son los siguientes:

- El usuario quiere cada vez más calidad en los recursos audiovisuales que se visualizan. Actualmente encontramos pantalla de resoluciones muy altas como por ejemplo los nuevos monitores 4K (cuya resolución es 4 veces mayor a Full HD).
- Internet cada vez está más presente en los equipos que utilizamos (televisión, móviles, relojes, etc.). Actualmente el 70,9% de la población Española cuenta con acceso a Internet por banda ancha en su casa. Además es importante destacar que el 47,2% de la población dispone de acceso a Internet a través de teléfono móvil [8].
- Con la integración de Internet cada vez en más ámbitos de nuestra vida se está aumentando el número de veces que la gente accede a la red diariamente. En 2007 ya un 44% de la población española accedía a Internet de forma regular, en 2013 este porcentaje ya era un 66%. Sobre todo la llegada de los Smartphones a sido la causa de este incremento rápido ya que nos permiten acceder a Internet en cualquier parte y en cualquier momento.



Porcentaje de personas que utilizan regularmente internet [9]

- Los propios hábitos y gustos de los usuarios hacen que una gran parte del tráfico que se demanda sea multimedia. Esto se acentúa aún más en dispositivos móviles. En Noviembre de 2013 el tráfico multimedia ya era de un 40% en ordenadores portátiles, 50% en tablets y 30% en smartphones. Además la llegada de la Voip IP para sustituir las llamadas convencionales acentuarán aún más estos porcentajes.



Volúmenes de tráfico por tipo de dispositivo [10]

- El movimiento de los grandes proveedores de datos a la nube ha cambiado el modelo tradicional de la de las redes. Antiguamente el envío de datos masivo se realizaba desde puntos muy localizados en la red. Ahora con la llegada de

soluciones como CDN, que replica los contenidos por distintos puntos de la red para reducir la distancia entre proveedor y cliente, los grandes proveedores de datos se encuentran en toda la red de forma distribuida.

El gran problema existente en que las previsiones de tráfico finalmente se cumplan y las redes no estén preparadas para ello es que se produzcan saturaciones en los servicios con mayor frecuencia.

Las saturaciones pueden ser aceptables para tráficos como el web que no requiere que el tráfico llegue de un punto a otro de forma rápida y sin variaciones, sin embargo, en tráfico como el multimedia las variaciones de retardos o los retardos prolongados crean problemas en el servicio.

Si además añadimos que, como se ha expuesto anteriormente, los servicios multimedia serán responsables de la mayoría de tráfico que se moverá por Internet, estaríamos ante un gran problema.

Además contamos con la condición añadida de que la solución para el problema no debe influir en la factura del usuario final que está acostumbrado a tener un servicio cada vez más barato y con más prestaciones.

4.2. Análisis del estado del arte

Actualmente, los operadores utilizan en sus redes tecnologías que permiten ofrecer servicios de conectividad que mantienen la independencia entre sus clientes y que permiten escalar en función del número de ellos.

4.2.1. Las redes actuales: basadas en MPLS

Este protocolo de encaminamiento es uno de los más usados en las redes de operadores para la definición de rutas entre origen y destino. Se basa en el uso de una serie de etiquetas que se insertan delante del paquete para que sea enrutado a través de los distintos nodos entre origen y destino.

Cuando un paquete llega a un nodo de la red, éste comprueba las etiquetas y consulta su tabla de routing en la que aparece la interfaz por la que será enviado el paquete. Además en algunos nodos también se procede a realizar un intercambio de etiquetas para definir el camino que se tomará en los siguientes nodos.

Hasta ahora MPLS había sido un protocolo suficiente para la gestión de tráfico en las redes actuales. La mayor ventaja que ofrece es la velocidad de enrutamiento que se obtiene a través del etiquetado de los paquetes. Cuando un switch tiene que enrutar un paquete solo tiene que consultar los primeros bits del paquete y consultar la interfaz de salida por la que enviar el paquete.

Sin embargo, poco a poco MPLS está dejando de ser un protocolo eficiente para el enrutamiento de tráfico ya que funciona de forma distribuida y es muy poco flexible. La principal carencia con la que cuenta MPLS es el tiempo de configuración que requiere cualquier cambio en la red. Al requerir configuraciones individuales de cada nodo, cualquier variación en la red conlleva un esfuerzo grande por parte del operador.

Además, la gran diferencia de fabricantes existentes que ofrecen dispositivos con soporte MPLS agrava el problema si nos encontramos en una red formada por diferentes marcas de switch. Cada marca funciona de una forma y requiere una configuración concreta para integrarse en la red.

4.2.2. VPNs, la necesidad de interconectar redes separadas entre sí

Hoy en día las empresas tienden a tener una estructura cada vez menos centralizada. Es muy común encontrar empresas en las que sus sedes se distancien cientos o miles de kilómetros. Estas empresas necesitan una forma que les permita comunicarse y compartir recursos de la forma más directa posible.

Las VPN permiten esta interconexión directa simulando que las sedes se encuentran unas al lado de otras y evitando las barreras que puede suponer que cada sede trabaje en su propia red aislada.

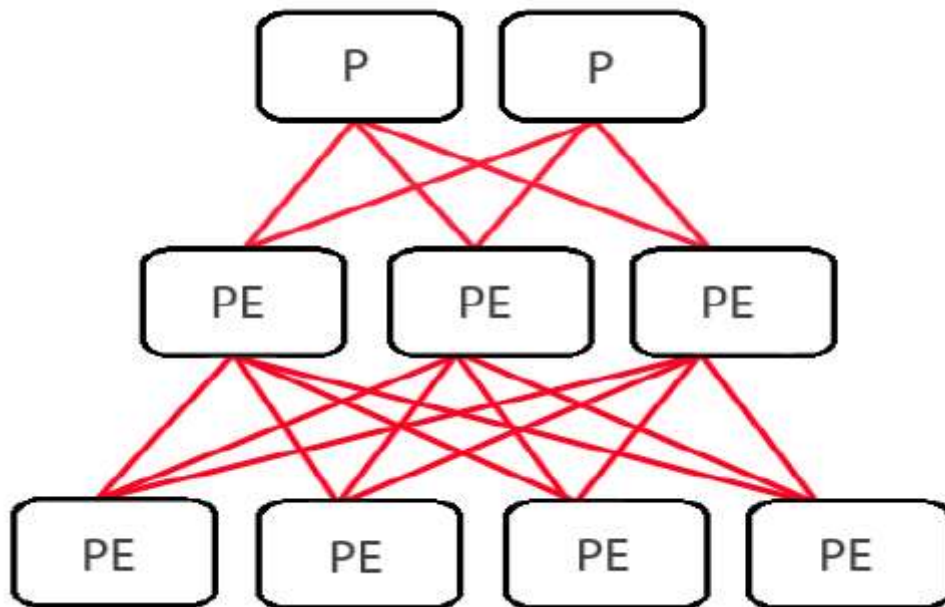
En el mercado actual podemos encontrar VPNs de nivel 2 y de nivel 3. La diferencia entre ellas es que la VPN de nivel 2 trabaja a nivel Ethernet, simulando que las dos redes se encuentran en una misma red local, y la de nivel 3 trabaja a nivel IP, estableciendo mayor independencia entre las redes.

4.2.3. EoMPLS: Ethernet sobre MPLS

Una vez explicado en que consiste el protocolo MPLS y las redes VPN nos encontramos con información suficiente para comprender el funcionamiento de las redes existentes en los operadores hoy en día.

En este punto se explicará el funcionamiento de EoMPLS, la tecnología que utilizan hoy en día muchos operadores para ofrecer servicios de VPNs a empresas. La tecnología está basada en su gran mayoría en el protocolo MPLS al que se añade una estructura jerárquica de red que permite reducir la posibilidad de saturaciones.

En EoMPLS la red está formada por una serie de nodos definidos como nodos P y nodos PE que se encuentran interconectados todos ellos entre sí a través de túneles. Se puede observar la forma en la que se distribuye la red en la imagen inferior.



Como se puede observar, se trata de una red jerárquica en la que se establece un primer nivel de PEs a los que se conectan los clientes, otro nivel de PEs que reducen la complejidad de la red y un nivel final de nodos P que actúan como nodos de tránsito.

Cuando un paquete llega a la red se envía de nodo PE a nodo PE hasta que se entrega en el destino. El paquete en su viaje por la red cuenta con dos etiquetas que definen por donde tiene que ir. En la parte interna se encuentra la etiqueta Pseudowire que solo es visible por los PEs. Encima de esta se introduce una etiqueta que identifica al túnel.



Al dar de alta un servicio es necesario configurar todos los PEs por los que pasarán los paquetes para llegar desde la sede origen a la sede destino. Además se establecen varios caminos de backup que sirven para asegurar la conexión entre nodos en todo momento. Aquí concretamente se encuentra uno de los problemas que están obligando a buscar nuevas soluciones que aceleren el tiempo de configuración de un servicio nuevo.

Finalmente, al dar de alta un servicio también se establece limitaciones (Rate limit o QoS) a través de un policer que limita estos parámetros de acceso a la red.

4.2.4. La alternativa: SDN

SDN puede ser la solución a las nuevas necesidades de los usuarios actuales. Cada vez existen más servicios en la nube y cada vez se necesita una mayor flexibilidad en las redes para ofrecer servicios de forma eficiente.

SDN es un nuevo paradigma que permite abstraer el gran nivel de complejidad que existe en el plano de control de las redes actuales a través de un sistema centralizado que gestiona y configura la red desde un mismo punto. Así, por ejemplo, si es necesario cualquier cambio en la red no tendremos que preocuparnos por el fabricante de los nodos de la red ni del lugar donde se encuentren.

Además esta solución permite que la red se adapte al usuario a través de APIs que el proveedor puede poner a disposición del cliente, normalmente a través de terceros. Con ello se consigue un servicio mucho más personalizado.

El triunfo de este paradigma cambiaría completamente el modo de trabajar de las redes de Internet, llevando el control y la gestión de los datos desde la parte del hardware (los Switches) al lado del software (el controlador) desde el que se tiene una visión mucho más completa de la red.

Muchas grandes empresas ya se encuentran investigando la tecnología SDN para ser aplicada en sus redes. En el caso de Google por ejemplo, ya ha probado este tipo de protocolos centralizados en sus datacenters para el movimiento de grandes cantidades de datos.

Parece ser que incluso los fabricantes de equipos de interconexión, que en principio no deberían ser partidarios del avance de este paradigma centralizado, están atentos a su evolución. En Marzo de 2014, durante la jornada SDN multifabricante organizada por

Telefónica, expusieron las ventajas que, bajo su punto de vista, serán las que hagan de SDN el protocolo de encaminamiento que destaque en unos años [11] [12].

4.2.5. Los beneficios de SDN [13] [14]

En el punto anterior se explicaba que SDN podría ser una solución para que los operadores den respuesta a lo que hemos venido exponiendo, es decir, a las necesidades actuales y futuras de los usuarios, sin embargo, ¿Cuáles son concretamente las ventajas que ofrece SDN?

En primer lugar, es importante destacar que en SDN se realiza una separación del hardware y el software de la red (plano de datos y plano de control). Esto conlleva varias ventajas:

- Es posible cambiar la red de forma modular. Si por ejemplo necesitamos una mayor velocidad de enrutamiento en los nodos, la lógica de la red seguiría siendo la misma y solo habría que cambiar el hardware.
- Hardware y software más sencillo. Se reduce mucho la complejidad de la red al dividir el plano de datos del plano de control. Además los costes de realizar cambios en la red se reducen al formarse de elementos más sencillos.
- Visión global de la red. Desde el controlador se obtiene una visión general de la red y se pueden crear protocolos mucho más eficientes que serán aplicados en el plano de datos.
- Redes mucho más flexibles. Se reduce mucho la complejidad de realizar cambios en la red. Únicamente con modificar el controlador toda la red cambia.

En segundo lugar, otra ventaja destacable es que SDN puede ser implementado en equipos de diferentes fabricantes. Protocolos como Openflow, basados en SDN, permiten a los operadores abstraerse del hardware que existe por debajo y trabajar con redes compuestas de equipos multimarca sin realizar configuraciones concretas para cada equipo.

Finalmente, el modelo SDN permite un tratamiento más granular del tráfico, permitiendo tomar decisiones dependiendo de muchos más criterios que lo que se hace actualmente. También podemos crear APIs que modifiquen la red dependiendo de las necesidades del momento. Todo esto permite ofrecer un servicio más personalizado y una experiencia de usuario mucho mejor.

4.3. Requisitos

Este cambio de tecnología supondría un gran esfuerzo por parte de los proveedores de redes españoles.

Cambiar de un modelo distribuido de red a un modelo centralizado cambiaría completamente la forma de trabajar de los operadores que tendrían que buscar nuevos modelos de negocio basados en SDN. Se estaría llevando el control de la red desde el hardware de los nodos de la red al software de un nodo central desde el que se dirigiría todo.

Para conseguir este cambio es vital que se comience a investigar la tecnología y se empiece a poner en práctica en redes reales. Es importante ir obteniendo estadísticas y datos del funcionamiento de SDN que permitan implantarlo de forma fiable cuando sea necesario.

Además la implementación tendría que realizarse de forma gradual. En una compañía con mayores de clientes todos sus servicios deben ser probados de forma estricta antes de ponerlos en manos de todos ellos. Por ello es importante ir aumentando los usuarios de este servicio según se asegure la estabilidad y funcionamiento del mismo.

De todo esto, se saca que el principal requisito es el tiempo, que debe ser suficiente como para implantar SDN de forma segura para los usuarios y tras la investigación necesaria que permita mantener un balance de beneficios suficiente.

Todos los operadores y grandes proveedores de datos, como Google, se encuentran actualmente investigando formas de implantar SDN en sus redes para aumentar la eficiencia.

4.4. Restricciones

4.4.1. Restricciones generales para los operadores

Las principales restricciones que existen actualmente para la implantación de SDN son:

- Madurez. La necesidad de investigación del protocolo y su mejora. SDN es un protocolo aún bastante joven (nació en 2003) y que aún no está implementado en muchos de los equipos existentes.
- El coste de adaptar las redes existentes a un nuevo protocolo que obligaría a formar a un gran número de profesionales.

- La gran dificultad que existe en una compañía de telecomunicaciones con millones de usuarios para realizar cambios. Ofrecer servicios a tanta gente obliga a que los servicios sean probados de forma intensiva antes de que se pongan en manos de la gente.

4.4.2. Los fabricantes de hardware de red

Sin duda el cambio de la lógica de las redes hacia puntos centralizados de la red sería un gran cambio en la filosofía de las redes convencionales. Se dejaría de invertir tanto dinero en la instalación de nodos convencionales con soporte para multitud de protocolos y se aumentaría la inversión en el nodo central y en la creación de lógicas más eficientes.

Este posicionamiento ha alertado a los fabricantes de hardware de red y les ha obligado a lanzar alternativas a SDN que les permita seguir manteniendo el cerebro de las redes en sus equipos.

Por ejemplo, hace unos meses Cisco informaba sobre un nuevo protocolo en el que se encontraba trabajando llamado OpFlex [15] [16]. El objetivo principal de este protocolo es ofrecer unos beneficios semejantes a los ofrecidos por SDN pero manteniendo la filosofía de red distribuida.

Como se puede observar, los fabricantes de hardware de red pueden ser los grandes perjudicados del cambio de las redes actuales a una filosofía centralizada. Posiblemente esta sea una gran barrera para el cambio ya que centrarán sus esfuerzos en ofrecer alternativas que les permita seguir asegurando su posición dominante en el mercado.

4.5. Marco regulador

Existe un gran debate en cuanto a si las redes deberían ser diferentes dependiendo de lo que pagues y los servicios que contrates o si no debería existir ninguna limitación.

Un gran número de usuarios defienden la *Neutralidad en la red* que exige una regulación que obligue a los operadores a tratar todo el tráfico de sus clientes por igual. Estos usuarios consideran que la personalización de las redes sería una forma de aumentar los beneficios por parte de los grandes operadores.

Sin embargo, en el caso de implantar una regulación se estaría limitando en gran manera la evolución y mejora de las redes ya que las compañías no estarían incentivadas a hacerlo.

El protocolo SDN se vería afectado en gran manera por una regulación de este tipo ya que su beneficio principal, que es la personalización de las redes al usuario, se eliminaría o limitaría.

5. Diseño de la solución técnica

5.1. Objetivos

Los objetivos que se buscan con las pruebas que se han realizado son:

- Probar de forma práctica los beneficios que ofrece SDN actualmente.
- Contemplar cual es el estado actual de la tecnología.
- Demostrar la sencillez que ofrece SDN a la hora de configurar grandes redes.

Para conseguirlo se diseñará un controlador que se irá mejorando durante el aumento de complejidad de las pruebas con el que se pretende ponerse en la piel de los desarrolladores y conocer SDN desde su punto de vista.

5.2. OpenFlow: La implementación de SDN

Durante este documento se ha hablado de lo que es SDN y cuáles son sus ventajas teóricas frente a las tecnologías existentes actualmente. Sin embargo, aún no se ha introducido cual es la solución práctica que implementa esta nueva forma de pensar las redes de los operadores.

Actualmente se están desarrollando muchas alternativas que implementen la filosofía centralizada de SDN en un protocolo real y que permita empezar a construir redes reales basadas en ella. Sin embargo, el protocolo que hasta la fecha ha conseguido destacar es OpenFlow, desarrollado en la Open Networking Foundation (ONF).

El objetivo de la ONF ha sido desde el primer momento ofrecer un protocolo que permita a los investigadores prepararse para el futuro y empezar a probar sus modelos de negocio.

5.3. ¿Cómo funciona OpenFlow? [17] [18] [19]

SDN es un paradigma que cambiaría la forma de trabajar de las redes actuales. Hasta ahora la mayoría de protocolos de encaminamiento han tenido una visión parcial de la red. La filosofía de protocolos como MPLS, usado actualmente en redes reales de operadores para la creación de VPNs, es funcionar de forma distribuida.

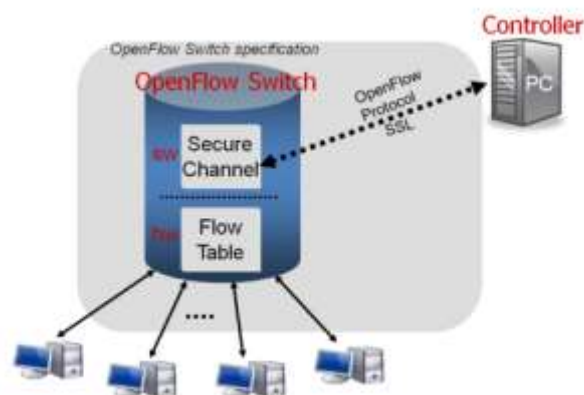
Sin embargo, SDN propone un modelo centralizado en el que un punto de la red cuenta con una visión completa de la red que gestiona y actúa en función del estado

de la red. Esta filosofía permite desarrollar protocolos más eficientes ya que es más fácil controlar todo lo que pasa en cada punto y modificar rápidamente el encaminamiento para evitar saturaciones y aprovechar mejor los recursos de la red.

En SDN se realiza una separación de la red en dos planos: plano de datos y plano de control.

El plano de datos es la parte de la red encargada de llevar los paquetes que entran por un router de borde hasta su destino. Este plano está gestionado por el plano de control que se encarga de decidir el enrutamiento que se realiza y contiene toda la lógica de la red.

La centralización del plano de control de la red se realiza en un punto llamado controlador. El controlador es el nodo de la red que cuenta con una visión global de la red. Es muy importante que el plano de control se encuentre lo suficientemente protegido, ya que sin él la red no podría funcionar. En redes que van a ser utilizadas por usuarios externos además los paquetes que envía y recibe el controlador son encriptados bajo protocolos como SSL para evitar que personas malintencionadas se puedan hacer con este tráfico o inyectar el suyo propio.



Esta filosofía de trabajo centralizado por la que apuesta SDN ya ha sido implementada en protocolos como OpenFlow. Este protocolo se ha posicionado como el protocolo basado en SDN predominante y actualmente se pueden encontrar implementaciones para bastante routes como por ejemplo el utilizado en este proyecto: Linksys WRT54GL.

En OpenFlow, las órdenes del controlador las reciben los Switch OpenFlow. Cada Switch cuenta con una tabla (Flow Table) que contiene las acciones que se toman cuando un paquete llega. La forma de trabajar es la siguiente:

- Un paquete llega al Switch por una interfaz.
- El protocolo SDN consulta la tabla de acciones y decide que cambios se deben hacer en el paquete.
- Realiza los cambios en el paquete y lo envía por las interfaces de salida.

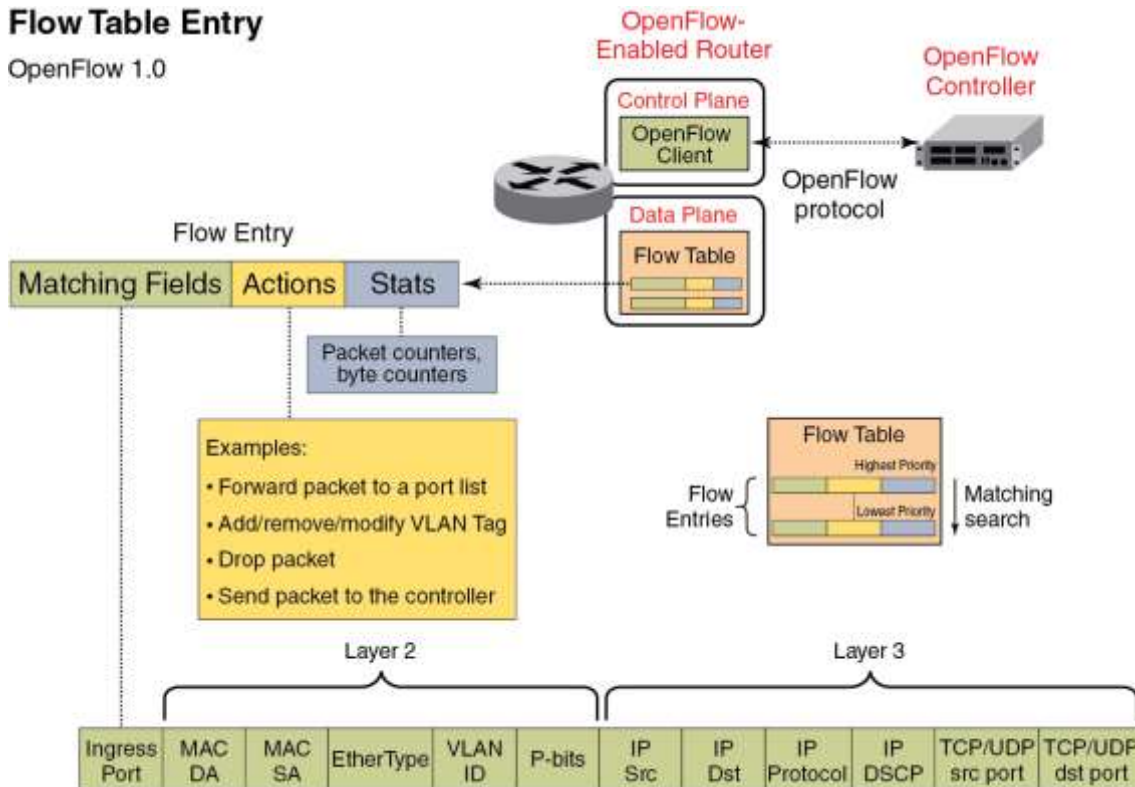
Es importante destacar que las reglas que se declaran en los Switch son omnidireccionales, es decir, lo que se aplica en un paquete que entra por la interfaz X y sale por la Y no tiene por qué ser igual que lo que se aplica si se entra por la Y y se sale por la X. OpenFlow permite crear reglas teniendo en cuenta un gran volumen de campos que residen en cada paquete que se mueve por la red:

Attribute	Meaning
in_port	Switch port number the packet arrived on
dl_src	Ethernet source address
dl_dst	Ethernet destination address
dl_vlan	VLAN ID
dl_vlan_pcp	VLAN priority
dl_type	Ethertype / length (e.g. 0x0800 = IPv4)
nw_tos	IP TOS/DS bits
nw_proto	IP protocol (e.g., 6 = TCP) or lower 6 bits of ARP opcode
nw_src	IP source address
nw_dst	IP destination address
tp_src	TCP/UDP source port
tp_dst	TCP/UDP destination port

Todos los parámetros de la tabla anterior, como ya se ha explicado, son guardados en una tabla de flujos (flow table). Esta tabla es utilizada para decidir el encaminamiento de cada paquete que llega a un nodo. La estructura de la flow table alojada en cada nodo de la red es la siguiente:

Flow Table Entry

OpenFlow 1.0



La tabla va dividida en tres partes principalmente:

- **Matching Fields:** Donde se alojan los datos que permiten diferenciar unos flujos de otros. Los valores de esta parte de la tabla se comparan con los campos de los paquetes para decidir que regla se corresponde con el paquete.
- **Actions:** Contiene el listado de acciones que deben realizarse en cada paquete antes de sacarlo por la interfaz de salida (que también viene representada en el campo Actions).
- **Stats:** Conjunto de contadores utilizados en algunas funciones del protocolo OpenFlow.

Esta tabla se va completando cuando el Switch OpenFlow recibe paquetes de control originarios del controlador. Cada paquete incluye una regla que se introduce en la flow table.

A continuación se explica el contenido de un paquete OpenFlow para la creación de una nueva regla, cuyo origen es el controlador de la red y cuyo destino es un Switch OpenFlow.

```
▼ Header
  Version: 0x01
  Type: Flow Mod (CSM) (14)
  Length: 80
  Transaction ID: 31
▼ Flow Modification
  ▼ Match
    ► Match Types
      Input Port: 1
      IPv4 DSCP: 0
      Protocol: IPv6 hop-by-hop option (0x00)
      IP Src Addr: 0.0.0.0 (0.0.0.0)
      IP Dst Addr: 0.0.0.0 (0.0.0.0)
      TCP/UDP Src Port: 0 (0)
      TCP/UDP Dst Port: 0 (0)
      Cookie: 0x0000000000000000
      Command: New flow (0)
      Idle Time (sec) Before Discarding: 0
      Max Time (sec) Before Discarding: 0
      Priority: 1
      Buffer ID: None
      Out Port (delete* only): None (not associated with a physical port)
    ► Flags
  ▼ Output Action(s)
    ▼ Action
      Type: Output to switch port (0)
      Len: 8
      Output port: 2
      Max Bytes to Send: 0
      # of Actions: 1
```

Paquete OpenFlow de modificación de flujo

Como se puede observar, el paquete se identifica bastante con los campos de la flow table de cada Switch.

- Versión: Identifica la versión de Openflow que se está utilizando. En este caso la versión 1.0.
- Type: Identifica el tipo de paquete que hemos recibido. En este caso un paquete tipo “Packet Out” que identifica a un paquete con las reglas a tomar en la salida del paquete.
- Packet Out: Esta sección contiene las reglas que se insertarán en la parte de la flow table “Matching Fields” como por ejemplo el puerto de entrada (Frame Recv Port) y las acciones que se tomarán que se corresponden con el bloque “Output Actions”.

5.4. Elementos utilizados

Las pruebas que se han realizado se dividen en dos bloques fundamentalmente:

5.4.1. Bloque 1

Pruebas en entorno virtual con redes de mayor envergadura. Para esta prueba se ha utilizado un emulador virtual de redes llamado Mininet que permite crear de forma instantánea entornos con soporte SDN.

El objetivo de este tipo de pruebas ha sido iniciarse de forma más rápida en el protocolo SDN y desarrollar controladores que podrán ser utilizados posteriormente entorno real sin necesidad de construir una gran infraestructura física.

Para crear una red virtual en mininet solo se requiere un archivo de configuración que identifique los nodos que formarán la red, la forma en la que están conectados y la tecnología que utilizará el controlador.

Como controlador para el entorno virtual se ha utilizado POX, un controlador de SDN que trabaja en lenguaje Python.

5.4.2. Bloque 2

Además de las pruebas en entorno virtual, se han realizado pruebas en entorno real con routers LinkSys WRT54GL. Para el montaje del escenario se han utilizado los siguientes elementos:

- Routers LinkSys WRT54GL como nodos de la red.
- Firmware Pantou (basado en OpenWRT) que implementa el protocolo SDN en los Linksys WRT54GL.
- Controlador Pox en un equipo con distribución Ubuntu instalada.

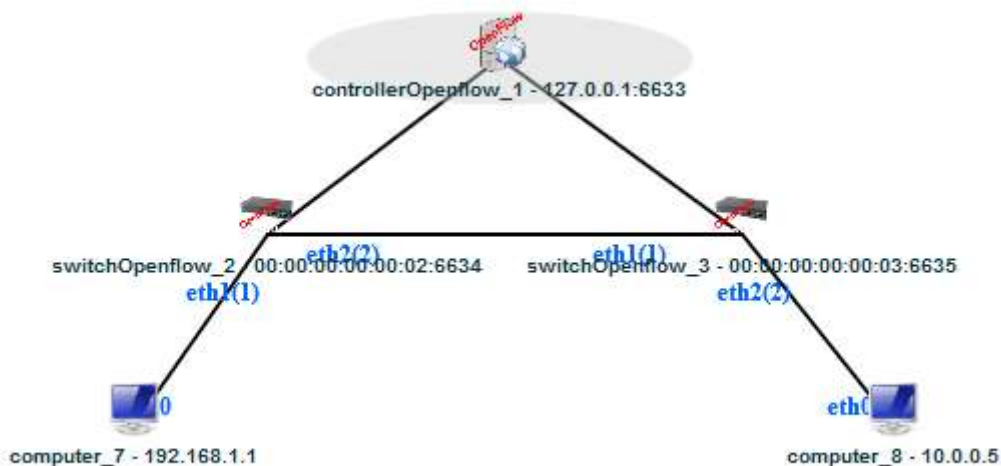
El objetivo de este bloque de pruebas es observar cómo sería implantar SDN en un entorno real y el esfuerzo que supondría el cambio.

Versión de WRT54GL	Velocidad de CPU	Capacidad de memoria RAM	Capacidad de memoria flash	Más información
1.0	200 MHz	16 MB	4 MB	Se lanzó en el año 2005 para soportar firmware de terceros, esencialmente es igual al WRT54GS v4.0 en cuanto a hardware. No viene con SpeedBooster, sin embargo se puede tener esta característica con firmwares de terceros. Sus números de serie comienzan con CL7A.
1.1	200 MHz	16 MB	4 MB	Chipset Broadcom BCM5352 . CPU MIPS a 200 MHz. Antenas extraíbles. Firmware de origen basado en Linux , fácilmente sustituible por firmware de terceros como OpenWrt [3] , DD-WRT [4] o Tomato [5] , entre otros. Básicamente igual a la versión 1.0. Sus números de serie comienzan con CL7B / CL7C / CF7C.

5.5. Estructuras de red

Durante la fase de pruebas se han utilizado 3 tipos de red diferentes en las que se ha ido aumentando la complejidad. A continuación se realizará una pequeña introducción de las redes que se han utilizado y para que se ha utilizado cada una:

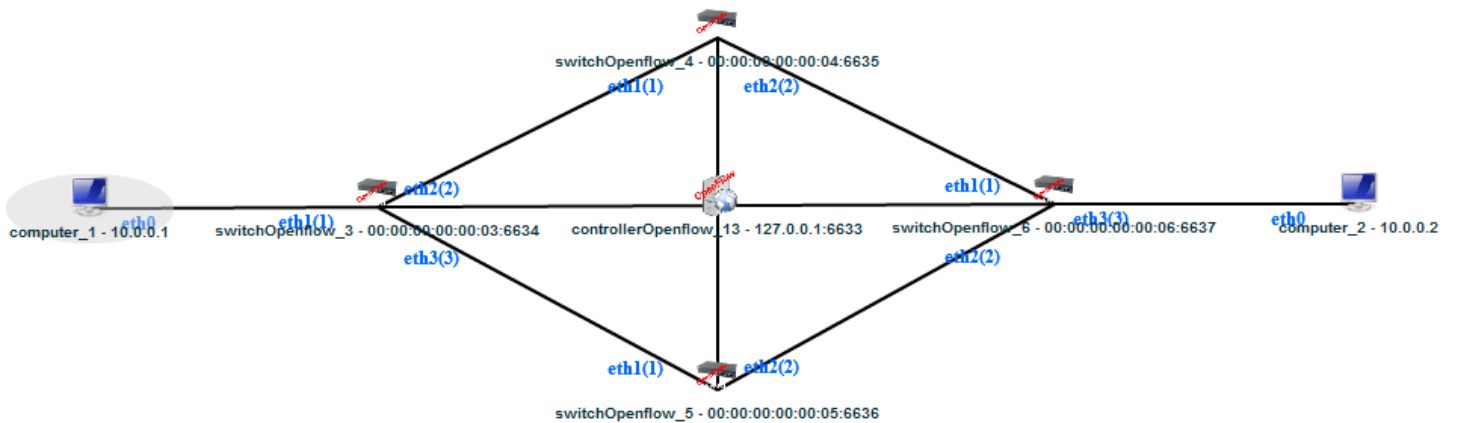
Red número 1 – Red simple



Esta red está formada por 2 equipos, 2 Switch OpenFlow y un controlador. Esta estructura ha sido utilizada para realizar la primera toma de contacto con el protocolo SDN tanto en el entorno virtual como en el entorno real.

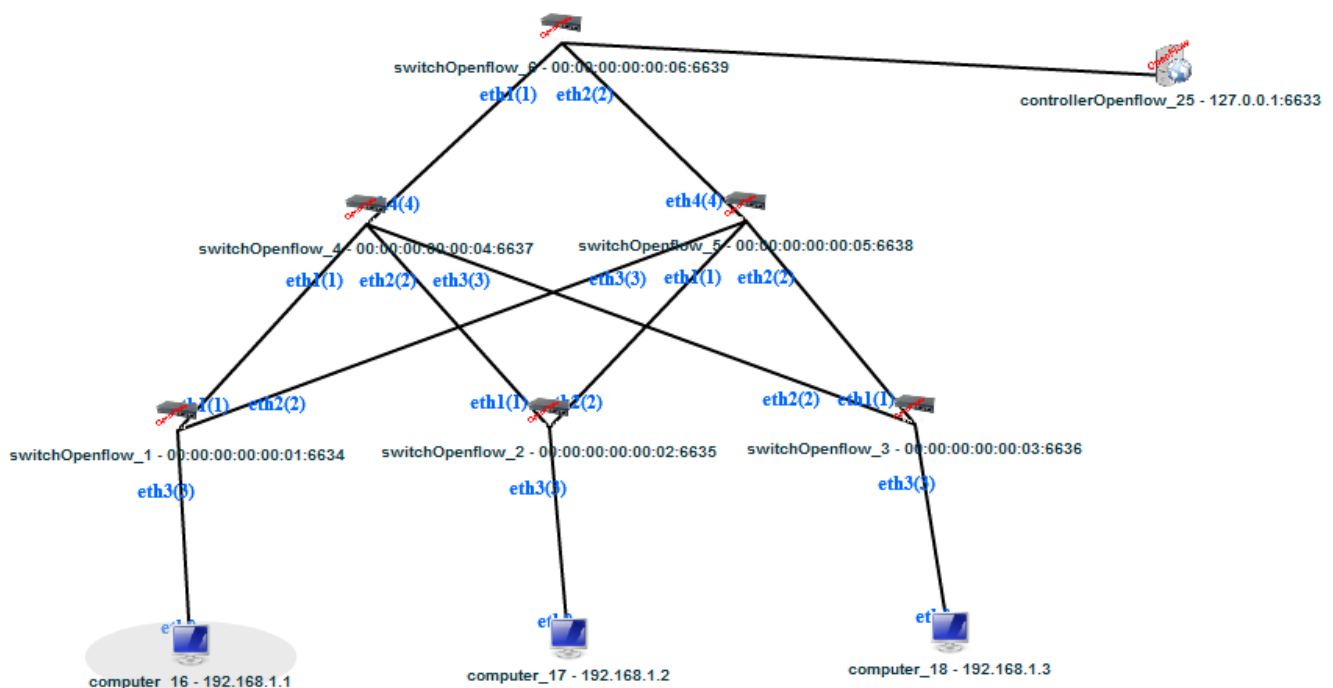
En el caso del entorno real el Switch es un router LinkSys WRT54GL y el controlador se ha incluido en uno de los equipos que se quieren interconectar.

Red número 2 – Red con camino de backup



Con esta estructura de red se ha estudiado la forma de crear caminos de backup (alternativos) en redes SDN. Con ayuda de esta red se ha estudiado la API que ofrece el controlador Pox para el manejo de paquetes y la detección del estado de los enlaces.

Red número 3 – Red jerárquica



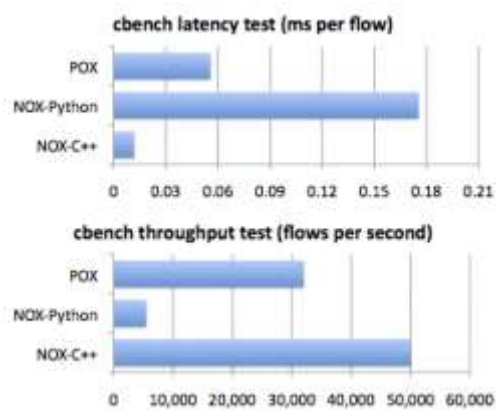
Para los últimos tests se ha utilizado una red basada en la estructura que actualmente se utiliza en operadores reales. Se trata de una red formada por niveles jerárquicos en la que el tráfico se dirige a los switch superiores solo si no existe ningún otro camino de interconexión hacia niveles inferiores.

Para conseguir que esta red funcione de forma satisfactoria se ha creado un controlador que detecta caídas de enlaces y activa y desactiva los flujos a partir del estado de las conexiones y las prioridades establecidas en los flujos que se pueden configurar.

En próximos apartados se explicará de forma más concreta el funcionamiento de esta red.

5.6. Características del controlador

Como se ha explicado anteriormente el controlador utilizado en este proyecto ha sido POX, un controlador programado en Python y basado en NOX, predecesor de POX cuyo código fuente está escrito en C++.



POX es un controlador de código abierto que cuenta con una amplia API que permite gestionar todos los Switch Openflow. Además, ya cuenta con algunos controladores que permiten realizar por ejemplo el Spanning Tree de la red para evitar bucles.

Una ventaja que ofrece es que, al estar escrito en Python, el controlador desarrollado podrá ser utilizado en cualquier sistema operativo [19].

5.6.1. Detección de estado de enlaces

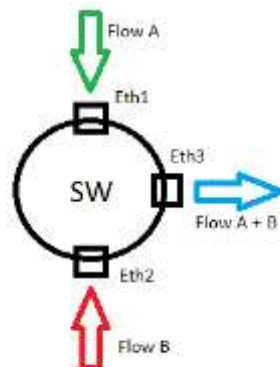
La detección del estado de los enlaces es algo muy importante en las redes actuales para realizar modificaciones en función de caídas o apariciones de enlaces.

En las pruebas de este proyecto se ha trabajado en el desarrollo de un controlador que detecta de forma automática las modificaciones de la red y activa y desactiva los flujos de tráfico en función de la disponibilidad de cada enlace y una serie de prioridades preestablecidas.

El funcionamiento final es muy similar al de MPLS cuando se establecen caminos de backup. Sin embargo la modificación de cualquier flujo o prioridad se puede realizar directamente desde el controlador, lo que reduce considerablemente el tiempo necesario para configurar cada nodo de la red.

5.6.2. Merging

Además en el controlador se ha establecido un algoritmo de merging mediante el cual se unen los flujos pertenecientes a una misma VPN cuando llegan paquetes a un Switch por varias interfaces de entrada y según las reglas SDN todos deben salir por la misma interfaz.



Como se puede observar en la imagen, a partir de los flujos A y B se obtiene un nuevo flujo con los dos flujos mezclados. Una vez mezclados los dos flujos ambos son tratados en el resto de la red de la misma forma. Esto en algunas ocasiones no es una solución válida ya que pueden quererse tratar de forma distinta forma los flujos (diferente reserva de recursos, diferentes prioridades, etc.).

5.6.3. Simplicidad en inserción de flujos

Una de las complejidades que se suelen identificar cuando se comienza a trabajar con SDN es la gran cantidad de reglas que se deben introducir una a una en los Switch OpenFlow.

En controladores de redes pequeñas es relativamente sencilla la inserción de reglas ya que se deben configurar pocos flujos. Sin embargo, en redes de gran tamaño la configuración de cada Switch puede ser bastante compleja ya que se le deben introducir bastantes reglas.

Por ello, en el controlador creado en este proyecto se han introducido algunas funciones que reducen notablemente la complejidad. En el controlador utilizado para las pruebas la configuración de los Switch se realiza flujo a flujo y no regla a regla como originariamente permite la API del controlador POX.

5.6.4. Estructura del código y reutilización del controlador

El controlador que se ha desarrollado para las pruebas ha sido estructurado de forma de que pueda ser utilizado por cualquier tipo de red únicamente modificando los flujos que se desean introducir en la red.

La estructura con la que se ha diseñado el código es la siguiente:

Importación de librerías
Declaración de los DPID* de los Switch
Declaración de los flujos de red y sus prioridades
Función "install_flows": Se ejecuta en el arranque de la red para realizar la primera configuración. Llama a select_best_flow(), preinstall_flow(), y install_flow() para realizar los pasos de configuración.
Función "select_best_flow": Selecciona de entre los flows que se le dan por parámetros cual es el que se usará. El flow seleccionado será el que mayor prioridad tenga y el que tenga todos los enlaces que utiliza activos.
Función "preinstall_flow": Crea las reglas necesarias para configurar los flows obtenidos en select_best_flow() y las añade a un array de reglas en el que se realiza la operación de merging.
Función "install_flow": Se envían las reglas a los Switchs

Además existen 2 escuchadores que permiten monitorizar el estado de los enlaces y actuar en consecuencia:

<code>_handle_ConnectionUp()</code>	Detecta cuando están conectados todos los Switch necesarios al controlador y realiza la instalación inicial de los flujos.
<code>_handle_PortStatus()</code>	Detecta cuando hay un cambio en un enlace (ADD/REMOVE) y reconfigura completamente la red con la infraestructura disponible.

Esta estructura simplifica mucho la tarea de configuración de la red. Además, al estar bien identificada cada fase de la configuración, es muy fácil realizar mejoras sobre el código del controlador

DPID: Número que identifica al Switch con el que el controlador se está comunicando.

5.7. Casos de testing

5.7.1. Caso 1: Red simple con 2 Switchs y 2 Hosts (Mininet) + Controlador Pox

Descripción del caso

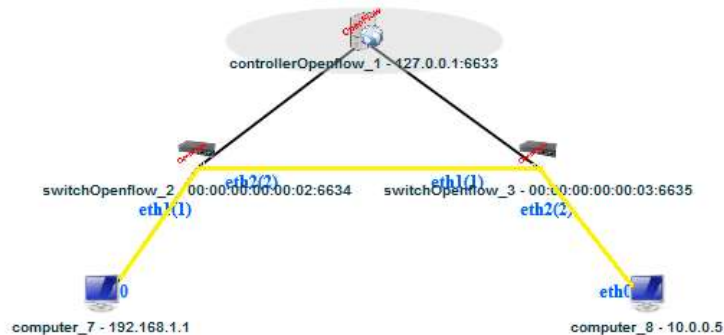
Este caso de prueba ha sido realizado trabajando con una red simple formada por 2 switchs y 2 hosts. El objetivo de estas pruebas ha sido la iniciación en la forma de trabajar de los siguientes elementos:

- Protocolo OpenFlow.
- Emulador de redes Mininet.
- Controlador POX y su API.

Además, este caso de prueba ha sido probado en un entorno real utilizando routers Linksys WRT54GL en los que se ha instalado Pantou, una distribución basada en OpenWRT con soporte de OpenFlow, y equipos con la distribución Ubuntu instalada y funcionando con el controlador POX.

La instalación que se ha realizado para que los switches Linksys WRT54GL funcionen dentro de la red SDN se describe en el punto 5.8 de esta memoria.

Flujos configurados



Flujo de conexión entre H7 y H8

```
mininet> dpctl dump-flows
*** s3 ***
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=94.047s, table=0, n_packets=4, n_bytes=280, idle_age=70, priority=1000,in_port=1 actions=output:2
cookie=0x0, duration=94.046s, table=0, n_packets=72, n_bytes=12616, idle_age=4, priority=1000,in_port=2 actions=output:1
*** s4 ***
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=94.016s, table=0, n_packets=72, n_bytes=12616, idle_age=4, priority=1000,in_port=1 actions=output:2
cookie=0x0, duration=94.054s, table=0, n_packets=4, n_bytes=280, idle_age=70, priority=1000,in_port=2 actions=output:1
```

Flujos iniciales configurados

NOTA: La configuración de los flujos puede consultarse en el Anexo 1 de esta memoria.

Trabajo a partir del caso de uso

Este caso se ha tomado como base para el aprendizaje del funcionamiento de SDN en el protocolo OpenFlow. La elección de una red pequeña para aprender el funcionamiento de los componentes de la red ha permitido una mayor agilidad en el resto de casos.

Las habilidades adquiridas a través de este caso han sido las siguientes:

- Redes OpenFlow virtualizadas utilizando Mininet: Esta ingeniosa herramienta permite generar redes Openflow sin necesidad de realizar una inversión en los elementos de la red. Este software es una potente herramienta incentiva a los desarrolladores a aportar mejoras a Openflow y POX sin tener que gastar

dinero para ello. En este caso de uso aprendí a crear un fichero de configuración que permite a Mininet generar una red completa.

- Uso básico de la API de POX: Pox es un controlador con una completa y extensa API que permite realizar controladores muy completos. En este caso de uso me he familiarizado con funcionalidades sencillas de Pox como la que permite enviar y borrar reglas en los Switchs y la obtención de parámetros importantes para realizar configuraciones (ej: DPID).
- Instalación de una red OpenFlow real: También fue posible crear el escenario del caso en un entorno real, lo que permitió conocer los tiempos que requiere configurar los distintos elementos de la red.
- Familiarización con Python: Pox es un controlador programado en Python y cuyos ficheros de configuración se generan en Python. Este caso me permitió familiarizarme con el lenguaje y darme cuenta de la agilidad de programación que ofrece y que le ha llevado a ser uno de los lenguajes punteros en la programación de scripts actualmente.

Resultados obtenidos

Como ya se exponía anteriormente, el resultado principal que se ha obtuvo con este caso es un aumento notable de la agilidad utilizando todos los elementos que configuran una red SDN.

Además, he podido comprobar una de las ventajas que hacen de SDN un paradigma interesante para la evolución de las redes actuales: el tiempo de instalación. En apenas dos horas, con el conocimiento adecuado y un controlador adecuado, puedes tener una red SDN sencilla funcionando.

Es muy notable el aumento de agilidad que se obtiene al separar el plano de control y el de datos. La tediosa tarea de generar configuraciones que son introducidas en cada switch para que la red funcione se simplifica en gran medida a través del controlador que gestiona todo su entorno de forma global.

5.7.2. Caso 2: Red media con 4 Switchs y 2 Hosts (Mininet) + Controlador Pox (camino de backup)

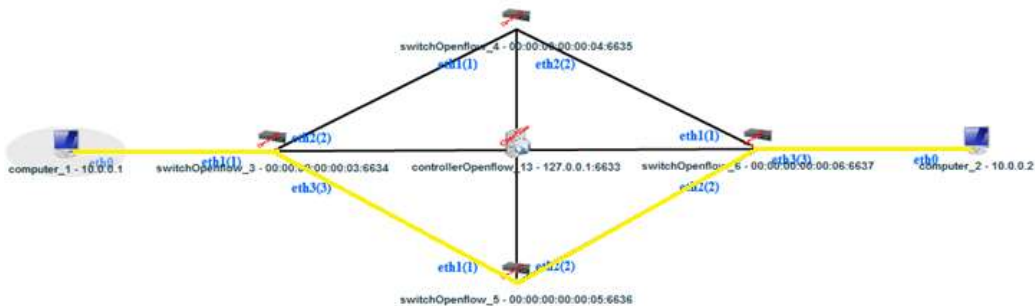
Descripción del caso

Este caso de prueba ha sido realizado trabajando con una red media en la que se han utilizado 4 Switches Openflow y 2 equipos entre los que se pretende establecer conexión.

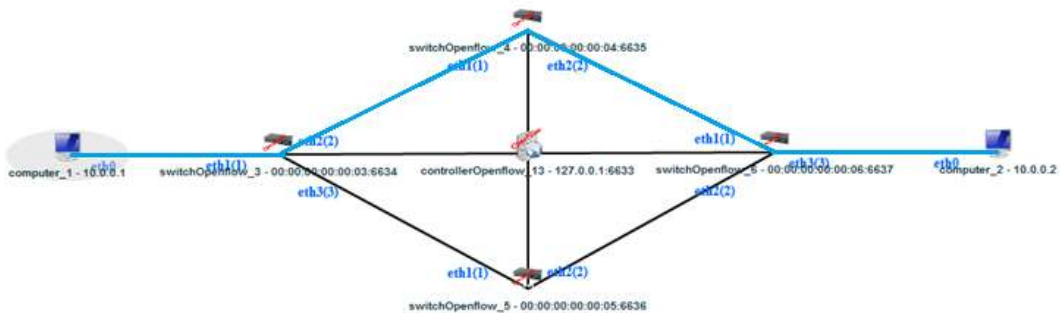
Esta vez la prueba se ha realizado únicamente en un entorno virtual con el que se ha buscado reducir el tiempo de montaje del entorno para generar una solución de mayor calidad.

En este caso se ha buscado solucionar a través de Openflow y Pox el problema que surge cuando se produce la caída de un enlace, estableciendo caminos alternativos por los que se llevan los paquetes.

Flujos configurados



Flujo de conexión entre H1 y H2 (Alta prioridad)



Flujo de conexión entre H1 y H2 (Baja prioridad)

```

mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.982s, table=0, n_packets=0, n_bytes=0, idle_age=7, priority=1,in_port=1 actions=output:2
  cookie=0x0, duration=7.981s, table=0, n_packets=10, n_bytes=1990, idle_age=0, priority=1,in_port=2 actions=output:
*** s2 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=7.995s, table=0, n_packets=8, n_bytes=1541, idle_age=1, priority=1,in_port=5 actions=output:4
  cookie=0x0, duration=7.996s, table=0, n_packets=0, n_bytes=0, idle_age=7, priority=1,in_port=4 actions=output:5
*** s3 -----
NXST_FLOW reply (xid=0x4):
*** s4 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=8.02s, table=0, n_packets=0, n_bytes=0, idle_age=8, priority=1,in_port=1 actions=output:2
  cookie=0x0, duration=8.02s, table=0, n_packets=7, n_bytes=1240, idle_age=2, priority=1,in_port=2 actions=output:1

```

Flujos iniciales configurados

NOTA: La configuración de los flujos puede consultarse en el Anexo 2 de esta memoria.

Trabajo a partir del caso de uso

En este caso se ha mejorado el controlador para que permita el uso de varios caminos entre orígenes y destinos iguales utilizando prioridades. Así se consigue redireccionar el tráfico por otro camino en el caso de que algún enlace se caiga.

Las habilidades que se han desarrollado durante este caso han sido las siguientes:

- Uso de funciones avanzadas de la API de Pox. En este caso es necesario realizar un seguimiento del estado de los enlaces de cada Switch. Para la realización de este caso se han investigado las funcionalidades de Pox que permiten realiza esta tarea.
- Necesidad de una estructura de código modular. Al requerir funcionalidades más avanzadas, ha sido necesario estructurar bien el código del controlador para permitir modificaciones sin necesidad de cambiar todo el controlador.

Resultados obtenidos

En este caso se vuelve a apreciar de nuevo la agilidad de configuración que se obtiene en redes SDN. La configuración de los Switchs fue la misma que en el caso anterior y solo fué necesario modificar el controlador para aumentar las funcionalidades de la red.

Además, se pone de manifiesto otra de las ventajas de SDN, la velocidad de reacción ante un cambio en la red.

En protocolos distribuidos como MPLS cuando se detecta un cambio en la red, la orden de cambio se va pasando nodo a nodo hasta que toda la red tiene conocimiento de ello. En SDN a caída del enlace se informa al controlador y desde allí se avisa a toda la red de los cambios a realizar.

Esta forma de trabajar es mucho más eficiente y reduce el tiempo a la siguiente suma de tiempos:

T. detección de la caída en Switch + T. envío de cambio al controlador + T. computo del algoritmo en el controlador + T. reconfiguración de flujos

Teniendo en cuenta que todos los tiempos que intervienen son muy pequeños, podríamos hablar de un tiempo despreciable siempre que el algoritmo de computo del controlador sea eficiente.

5.7.3. Caso 3: Red grande con 6 Switchs y 3 Hosts (Mininet) + Controlador Pox (camino de backup y merging)

Descripción del caso

Este caso de prueba ha sido realizado trabajando con una red media en la que se han utilizado 6 Switches Openflow y 3 equipos entre los que se pretende establecer conexión.

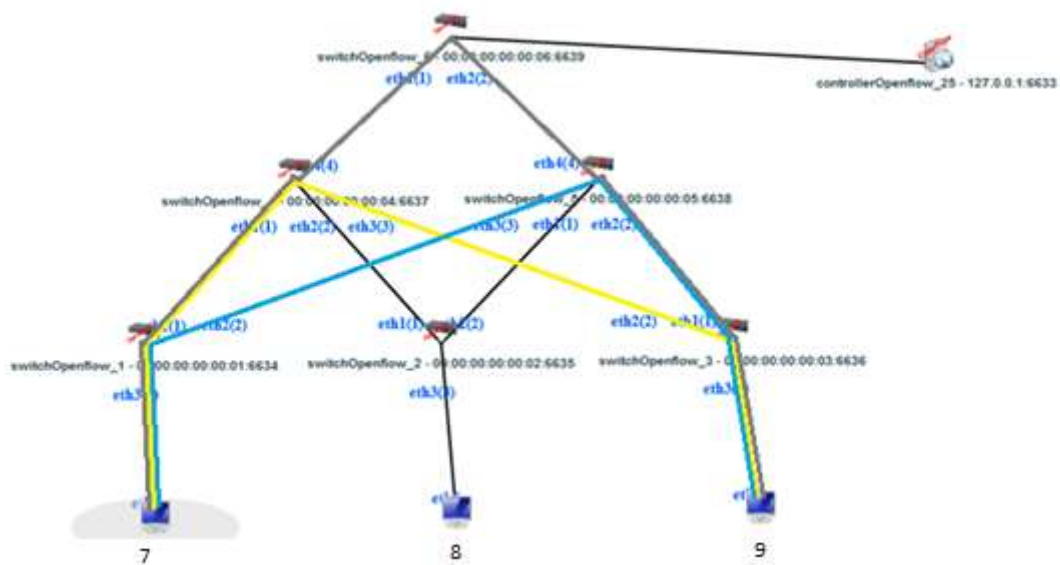
De nuevo, este caso de uso ha sido realizado bajo un entorno virtualizado, debido las dimensiones del mismo y a la mayor agilidad que ofrece un entorno virtual para realizar pruebas

Este caso de uso se ha desarrollado teniendo en cuenta dos factores principalmente:

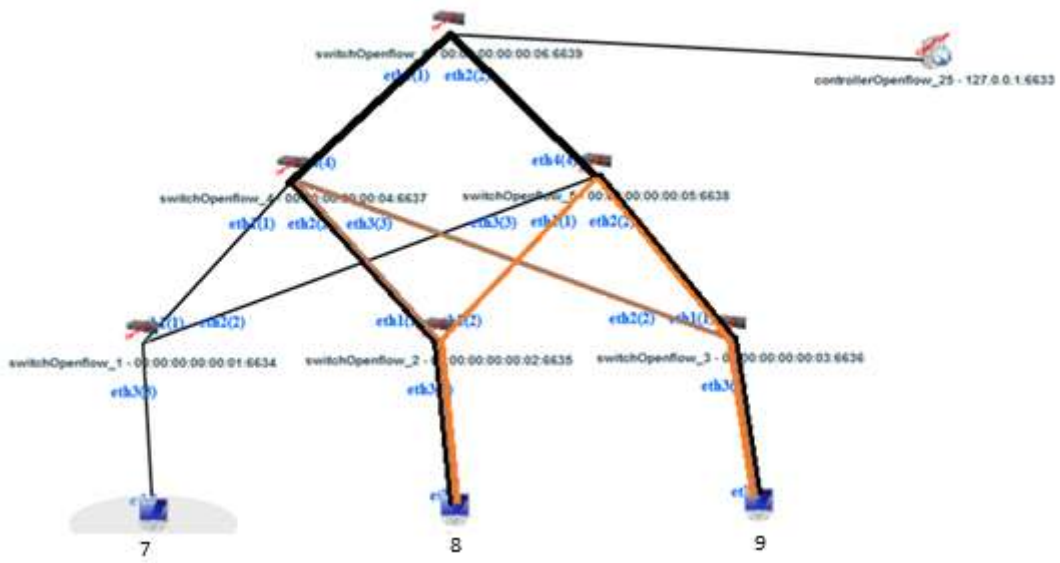
- La estructura de red se asemeja a una red real utilizada por los operadores actualmente para la configuración de VPNs entre sedes empresariales distanciadas entre sí. Se trata de una red jerárquica en la que los paquetes, que salen de los routers de borde se dirigen a los niveles superiores cuando no pueden ser transmitidos hacia el destino por el nivel en el que están.

- La necesidad de una estrategia de merging. En este caso se puede dar la situación de que dos paquetes que provienen de distintas interfaces de entrada quieren salir por la misma interfaz de salida. Cuando se trata de un escenario en el que solo existe una VPN, como en el que nos encontramos, los dos flujos de deben unir. Para hacer esto, ha sido necesario mejorar el controlador para soportar la funcionalidad.

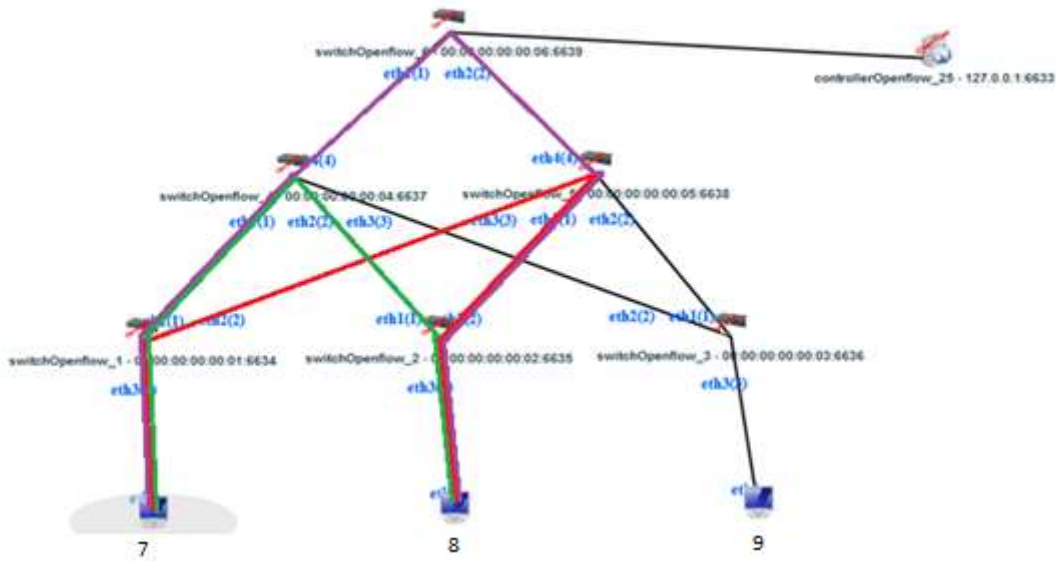
Flujos configurados*



Flujo de conexión entre H7 y H9



Flujo de conexión entre H8 y H9



Flujo de conexión entre H7 y H8

```

mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1.71s, table=0, n_packets=7, n_bytes=850, idle_age=0, priority=1,in_port=3 actions=output:1
 cookie=0x0, duration=1.71s, table=0, n_packets=0, n_bytes=0, idle_age=1, priority=1,in_port=1 actions=output:3
*** s2 -----
NXST_FLOW reply (xid=0x4):
*** s3 -----
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1.725s, table=0, n_packets=1, n_bytes=107, idle_age=0, priority=1,in_port=5 actions=output:4
 cookie=0x0, duration=1.725s, table=0, n_packets=4, n_bytes=600, idle_age=0, priority=1,in_port=4 actions=output:5
*** s4 -----
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1.757s, table=0, n_packets=3, n_bytes=510, idle_age=0, priority=1,in_port=3 actions=output:1
 cookie=0x0, duration=1.752s, table=0, n_packets=0, n_bytes=0, idle_age=1, priority=1,in_port=1 actions=output:3

```

Flujos iniciales configurados

NOTA: La configuración de los flujos puede consultarse en el Anexo 3 de esta memoria.

Trabajo a partir del caso de uso

En este caso de uso ha sido especialmente importante la estructuración del código de manera que sea fácil editarlo y entenderlo.

A través de este caso se ha podido apreciar también algunas carencias que actualmente tiene Pox y que conllevan una mayor complejidad del código del controlador.

Además se ha aprendido el funcionamiento de una red similar a la que usan actualmente los operadores para enrutamiento de tráfico y se ha estudiado la viabilidad de integrar SDN en ella.

Resultados obtenidos

Los resultados obtenidos han sido muy satisfactorios ya que se ha conseguido reproducir parcialmente el funcionamiento de una red jerárquica. El controlador podría ser mejorado para adaptarse en mayor medida a una red real en próximas investigaciones. En el apartado 8. Conclusiones se explican algunas líneas de investigación que permitirían incluso mejorar el funcionamiento de las redes actuales permitiendo cosas como desactivar nodos cuando no sean utilizados.

Además, durante este caso, se ha puesto en manifiesto la juventud del controlador Pox. Se ha detectado la carencia de unir realizar merging en el controlador de forma nativa a partir de su API. Esto llevo a realizar un controlador especial para este tipo de escenarios.

El resultado final tras los tres casos ha sido un controlador que ha evolucionado desde el caso 1 al caso 3 detectando las necesidades reales que tienen actualmente las redes de los operadores. Además se ha desarrollado con una estructura fácil de comprender y que permite que el controlador se reutilice en cualquier red OpenFlow que utilice Pox como controlador.

5.8. Descarga de controlador

El controlador desarrollado durante los casos de test de este trabajo está disponible para su descarga en el siguiente repositorio de GitHub:

<https://github.com/jpruden92/openflow-pox>

Para su descarga es necesario un cliente Git e introducir el siguiente comando en un terminal de comandos:

```
git clone https://github.com/jpruden92/openflow-pox.git
```

5.9. Configuración de router Linksys WRT54GL [20] [21]

Para la realización de las pruebas en un entorno real ha sido necesario realizar una configuración de los routers Linksys WRT54GL que han formado parte como nodos de la red. En todos los nodos de la red se ha instalado una distribución de Linux con soporte de Openflow llamada Pantou.

Pantou es una versión modificada del firmware OpenWRT, una distribución de Linux desarrollada específicamente para equipos Linksys WRT54GL y que permite extender de forma notable las funcionalidades del router. Pantou incorpora a OpenWRT varios procesos que le permiten enviar y recibir paquetes OpenFlow.

Para realizar la configuración de cada nodo de la red se ha seguido un proceso sencillo que ha permitido obtener en poco tiempo una red lista para funcionar. Esto es una gran ventaja si lo comparamos con las redes basadas en protocolos distribuidos en los que se incrementa el tiempo de configuración.

Los pasos seguidos son los siguientes:

1. En primer lugar debemos activar el router Linksys WRT54GL en modo Fail Safe. Para ello desconectamos el equipo de la corriente y lo conectamos hasta que el led DMZ se encienda. En este momento pulsamos el botón Reset hasta que el led DMZ comience a parpadear. Además configuramos el PC con una IP del rango 192.168.1.0/24.

2. Descargamos OpenWRT en nuestro equipo. Realizamos los siguientes comandos que descargarán OpenWRT en la carpeta tmp de nuestro ordenador:

```
cd /tmp
wget http://downloads.openwrt.org/whiterussian/newest/default/openwrt-brcm-2.4-squashfs.trx
wget http://downloads.openwrt.org/people/nbd/nvram-clean.sh
```

3. Enviamos OpenWRT al router. Para ello utilizamos el comando SCP que permite enviar ficheros de un equipo a otro:

```
cd tmp
scp jprudencio@192.168.1.130:/tmp/openwrt-brcm-2.4-squashfs.trx ./
mtd -r write openwrt-brcm-2.4-squashfs.trx linux
```

4. Descargamos e instalamos Pantou. De la misma forma que se instaló OpenWRT anteriormente, instamos ahora Pantou:

```
cd /tmp
wget archive.openflow.org/downloads/openwrt-images/brcm47xx/ofwrt-54g-rc1-squashfs.bin

cd /tmp
scp jprudencio@192.168.1.130:/tmp/ofwrt-54g-rc1-squashfs.bin ./
mtd -r write ofwrt-54g-rc1-squashfs.bin linux
```

5. Accedo por Telnet al router a través de la IP predefinida 192.168.1.1 en eth0 y ejecuto los siguientes comandos que deberán informarnos que los procesos que trabajan con OpenFlow están activados:

```
ps aux | grep ofprotocol
ps aux | grep ofdatapath
```

Una vez realizados estos sencillos pasos, ya dispondremos de un Linksys WRT54GL preparado para ser parte de una red gestionada a través del protocolo OpenFlow.

Además de este proceso de configuración, es importante para cualquier trabajo posterior con el controlador POX de OpenFlow conocer la forma de obtener el DPID de cada nodo ya que será el que nos permita identificar y enviar reglas a cada uno de ellos.

Para obtenerlo en POX nos podemos dirigir al atributo “dpid” del objeto “event” que se recibe cada vez que un nodo se conecta al controlador. Así por ejemplo, cada vez que se ejecute el evento de conexión de un nodo podemos obtener el DPID de la siguiente forma:

```
log.info(event.dpid)
```

6. Resultados y evaluación

6.1. Ventajas detectadas

A lo largo de la realización de este proyecto he detectado ventajas importantes frente a otras tecnologías que funcionan de forma distribuida.

6.1.1. Tiempo de adaptación

La principal ventaja, y que en mi opinión es la más importante, es el reducido tiempo de adaptación que requiere OpenFlow para empezar a trabajar con él. La separación del plano de control y el de datos permite que los desarrolladores de controladores se abstraigan de lo que pasa en el plano de datos y puedan desarrollar algoritmos de encaminamiento de forma mucho más rápida.

Con la filosofía SDN la red se puede mejorar de forma modular realizando investigación por una parte en el plano de datos, para conseguir equipos con una mayor capacidad de cómputo y análisis de paquetes, y por otra en el plano de control, obteniendo controladores rápidos y eficientes.

En los protocolos distribuidos la construcción de redes con switches de varias marcas puede aumentar mucho la complejidad de configuración debido a posibles problemas de compatibilidad. Abstraernos del equipo que tenemos debajo del protocolo OpenFlow es una gran evolución para que las empresas por fin no se encuentren limitadas a la hora de incluir tecnología de diversos fabricantes y realizar cambios rápidos.

Personalmente, la evolución que he seguido en el aprendizaje del funcionamiento de SDN y los protocolos que lo implementan como Openflow ha sido exponencial. En apenas 6 meses he conseguido comprender su funcionamiento y desarrollar redes dirigidas por POX.

La mayor dificultad que he encontrado estuvo en la comprensión de la API de POX y el proceso de adaptación a la misma. Como toda tecnología nueva, requiere un tiempo de introducción aprendizaje de los conceptos básico.

6.1.2. Visión general de la red

Tener una visión general de todo lo que pasa en la red aporta múltiples ventajas que hacen que SDN sea una tecnología mucho más eficiente que sus predecesores.

En primer lugar, saber en todo momento el estado de la red desde un punto central permite reaccionar rápidamente a problemas y saturaciones de enlaces.

En protocolos como MPLS, cuando un enlace falla se debe informar a toda la red de forma distribuida de la caída, pasando nodo por nodo y aumentando el tiempo de reconfiguración si nos encontramos en una red grande. En OpenFlow la caída se comunica directamente al controlador y éste comunica los cambios en la red a todos los switch rápidamente.

Además, en SDN los algoritmos de encaminamiento se ejecutan en el controlador y se comunican los resultados a todos los enlaces. En MPLS los algoritmos de encaminamiento ante una caída se ejecutan entre los nodos de la red.

En resumen, una filosofía centralizada como es SDN nos ofrece un control total y rápido sobre la red que aumenta la eficiencia sobre todo en momentos de saturación.

6.1.3. Redes de reducido presupuesto

Con SDN ya no es necesario realizar una gran inversión en Switchs con soporte de múltiples protocolos de encaminamiento. Posteriormente se explicará en el presupuesto, pero con poco dinero cualquier persona puede construir su red SDN y realizar encaminamiento de sus paquetes.

Además, al se OpenFlow un protocolo de código abierto, cualquiera puede aportar y obtener actualizaciones del protocolo. Esto permite que las funcionalidades de nuestras redes aumenten sin necesidad de realizar ninguna inversión.

6.2. Desventajas detectadas

Aunque SDN ofrece grandes ventajas frente a los protocolos que se utilizan actualmente, también es importante detallar algunas desventajas derivadas del trabajo con Openflow durante el desarrollo de este proyecto.

6.2.1. Mayor sensibilidad frente a ataques

Aunque los protocolos centralizados ofrecen grandes ventajas frente a los distribuidos, no debemos olvidar que estamos centralizando toda la gestión de la red en un punto concreto.

Esto hace que nuestra infraestructura sea más sensible a ataques contra el plano de control de la red o problemas en él. Una caída en el controlador de la red podría ocasionar la pérdida de servicio en miles de usuarios en el caso de los grandes operadores.

Para reducir este tipo de problemas, posiblemente sea necesario replicar los controladores de forma que si uno es atacado, el resto puedan sustituirle.

6.2.2. Protocolo muy joven

OpenFlow aún es un protocolo muy joven que muchos fabricantes no soportan. Además, los controladores que lo soportan se encuentran aún en proceso de prueba y de mejora.

Si tomamos como ejemplo el controlador utilizado en este proyecto, POX, nos encontramos la barrera de que aún no existen controladores lo suficientemente evolucionados como para ser utilizados en una red real.

Por este motivo, en este proyecto se decidió empezar a desarrollar un controlador personalizado que se adaptara a las necesidades de los casos de prueba realizados y que fuera fácil de comprender y de mejorar en el futuro.

Posiblemente, con el paso del tiempo OpenFlow y POX se convertirán en herramientas sólidas y fiables para el uso en entornos empresariales.

6.2.3. Necesidad de análisis de paquetes

En SDN no nos olvidamos de la necesidad de leer las cabeceras de un paquete para enrutarlo. Cada vez que llega un paquete a un switch se debe analizar para buscar la regla de flujo que le corresponde.

Este problema tendrá que ser sufragado invirtiendo en switches con gran capacidad de análisis de paquetes.

6.3. SDN: ¿Un candidato para el cambio?

No debemos olvidar, que para que SDN pueda ser un candidato para resolver los problemas del futuro debe cumplir algunas condiciones que se han ido exponiendo a lo largo de esta memoria.

Esto es importante, ya que, una solución resolverá verdaderamente un problema si se adapta perfectamente a las necesidades y requisitos del mercado que pretende cambiar.

A continuación se realizará un análisis de los requisitos que debe cumplir la posible solución del futuro y estudiaremos, a través de los datos obtenidos en las pruebas si esto se cumple.

- Servicio flexible: la solución elegida debe ser lo suficientemente flexible como para admitir cambios rápidos en la red cuando sean necesarios (añadir nuevos clientes, desviar tráfico cuando sea necesario, etc.)
 - o SDN cumple este requisito ya que al separar el plano de control y de datos se mejora mucho la capacidad de cambio de la red. Con cambios rápidos en el controlador podemos redirigir flujos y crear nuevos. Además escalar la red a una más grande es mucho más sencillo, permitiendo añadir nodos nuevos sin necesidad de cambiar el controlador.
- No romper la tendencia de precios del servicio: el precio de acceso a internet debe seguir la tendencia que se ha cumplido hasta el momento, precios cada vez más bajos y mejor servicio.
 - o SDN cumple este requisito, ya que permite una inversión continua y progresiva en mejora de la infraestructura sin necesidad de comprar nodos de interconexión nuevos para la red. Solo es necesario trabajar en una implementación de protocolos como OpenFlow en los nodos ya existentes. En este proyecto tenemos un ejemplo claro de esto, ya que se ha trabajado con un switch de bajo precio y cuyas capacidades se han potenciado mediante OpenFlow.
- Cumplir las necesidades del usuario: el futuro seguramente esté en los servicios personalizados al usuario. El cliente quiere pagar por lo que realmente necesita y que el servicio que contrata se adapte a él y no al revés.

- La posibilidad de movilizar la parte de control de la red al Software permite desarrollar redes mucho más dinámicas y mejor adaptadas a los clientes. En las redes distribuidas cada nodo debe ser configurado de forma unitaria, lo que hace muy costoso adaptar el servicio a cada usuario. En redes centralizadas, en la que se tiene una visión global de la red, realizar modificaciones que modifiquen los servicios del usuario es mucho más fácil.
- Tecnología suficiente: además de ser una solución teóricamente válida, ¿es también válida en la práctica?
 - Sin duda, este es uno de los puntos en los que SDN puede tener carencias. Como se ha demostrado en las pruebas, la filosofía centralizada por la que apuesta SDN es posible, sin embargo, hace falta investigación que permita aprovechar eficientemente la tecnología existente en las redes. Lo que sí hay que tener en cuenta es la reducción del tiempo necesario para implementar el plano de control, factor determinante para animar a desarrolladores independientes y empresas a investigar.
- Soporte de gran cantidad de tráfico: no olvidemos que el principal motivo de esta evolución es soportar de forma más eficiente un gran volumen de tráfico.
 - La ventaja de que protocolos como Openflow puedan implementarse en equipos con los que ya cuenta la empresa, es que el soporte de tráfico seguirá siendo el mismo. Con controladores eficientes lo que se consigue es que la red que antes era rígida y difícil de configurar, ahora sea dinámica y se adapte a las necesidades de cada momento.

7. Presupuesto y planificación del trabajo

7.1. Presupuesto

Como se ha explicado anteriormente, una de las principales ventajas que se han detectado durante el desarrollo del proyecto es que se pueden crear redes basadas en SDN con un reducido presupuesto. Las redes construidas durante este proyecto no sobrepasaran ninguna los 500 euros de presupuesto.

En la siguiente tabla se muestra el precio de cada componente unitario. Se corresponden con los utilizados en el proyecto:

Item	Precio	Enlace
Switch Linksys WRT54GL	49,99 €	http://www.amazon.es/Linksys-WRT54GL-EU-antenas-externas-puertos/dp/B000BZURG6
50 m de cable Ethernet	21,84 €	http://www.amazon.es/K%C3%B6nig-CMP-UTP6R50-Rollo-categor%C3%ADa-50/dp/B00652MRMQ
Conectores RJ45	5,62 €	http://www.amazon.es/K%C3%B6nig-CMP-RJ45PL-C6-Conector-RJ45-CAT-6/dp/B006RE5Y0E
Accesorio para cables RJ45	11,89 €	http://www.amazon.es/Konig-CMP-MODCRIMPL1-arrugador-cortador-separador/dp/B003WV55IG

A partir de esta tabla, se pueden generar los costes de cada red que se ha construido en el proyecto.

Red Pequeña	
2 Switchs Linksys WRT54GL	99,98 €
Cable Ethernet + Conectores + Accesorio	39,35 €
	Total: 139,33 €

Red Media	
4 Switchs Linksys WRT54GL	199,96 €
Cable Ethernet + Conectores + Accesorio	39,35 €
	Total: 239,31 €

Red Grande	
6 Switchs Linksys WRT54GL	299,94 €
Cable Ethernet + Conectores + Accesorio	39,35 €
	Total: 339,29 €

7.2. Planificación

Estructura de trabajo

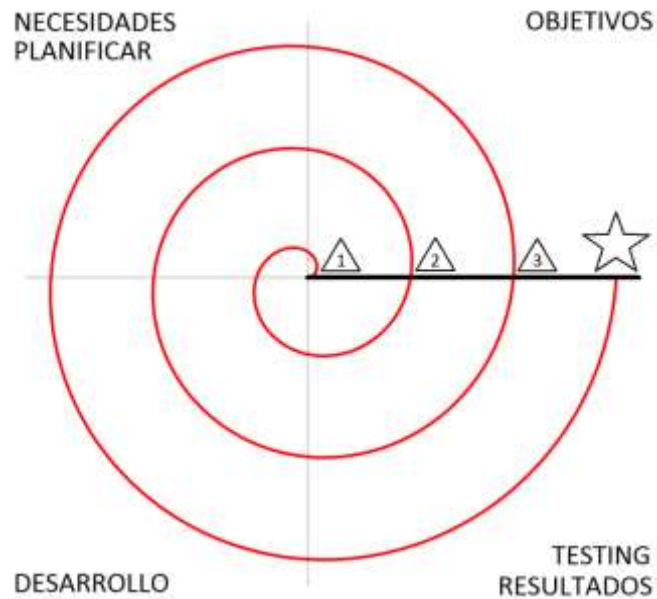
Para el desarrollo del proyecto se han seguido una serie de fases muy diferenciadas con las que se ha buscado una evolución gradual en el aprendizaje del paradigma SDN y los elementos que lo aplican de forma práctica (Openflow y Pox).

La estructura de fases que se ha seguido ha sido la siguiente:

- Introducción en protocolos de encaminamiento centralizados basados en SDN.
- Estudio del protocolo Openflow y controlador Pox para construir una red basada en SDN.
- Construcción de caso de uso 1.
- Construcción de caso de uso 2.
- Construcción de caso de uso 3.
- Recopilación de resultados y datos.
- Confección de memoria.

Como se puede apreciar, el proyecto ha contado con partes teóricas en las que se ha estudiado la tecnología que se iba a utilizar (inicio) y se ha llevado a cabo un análisis de los resultados, y partes prácticas durante las que se ha restreado lo aprendido.

Es interesante detallar que en la realización de los casos de uso se ha seguido una planificación en espiral. La forma de trabajar y de obtener el controlador final ha sido la siguiente:



Como se puede observar la línea de trabajo (marcada en color rojo) pasa por cuatro fases que son iguales en cada giro:

- Objetivos: En esta fase se plantean los objetivos que se pretenden obtener en el caso en el que nos encontramos.
- Análisis de necesidades y planificación: Se buscan las necesidades para que permitirán cumplir con el objetivo y se planifica el trabajo de desarrollo.
- Desarrollo: Se genera la solución que cumpla con los objetivos.
- Testing y resultados: Se prueba la solución y se analizan los resultados.

Al final de cada iteración de la espiral se obtiene un prototipo. Esto me ha permitido planificar el trabajo a realizar en hitos tangibles, consiguiendo en cada iteración un controlador que puede ser probado.

El inicio de cada caso está señalado con los números 1 a 3. Además, la obtención del prototipo final ha sido marcada al final de la espiral con una estrella.

Fases y sub-fases

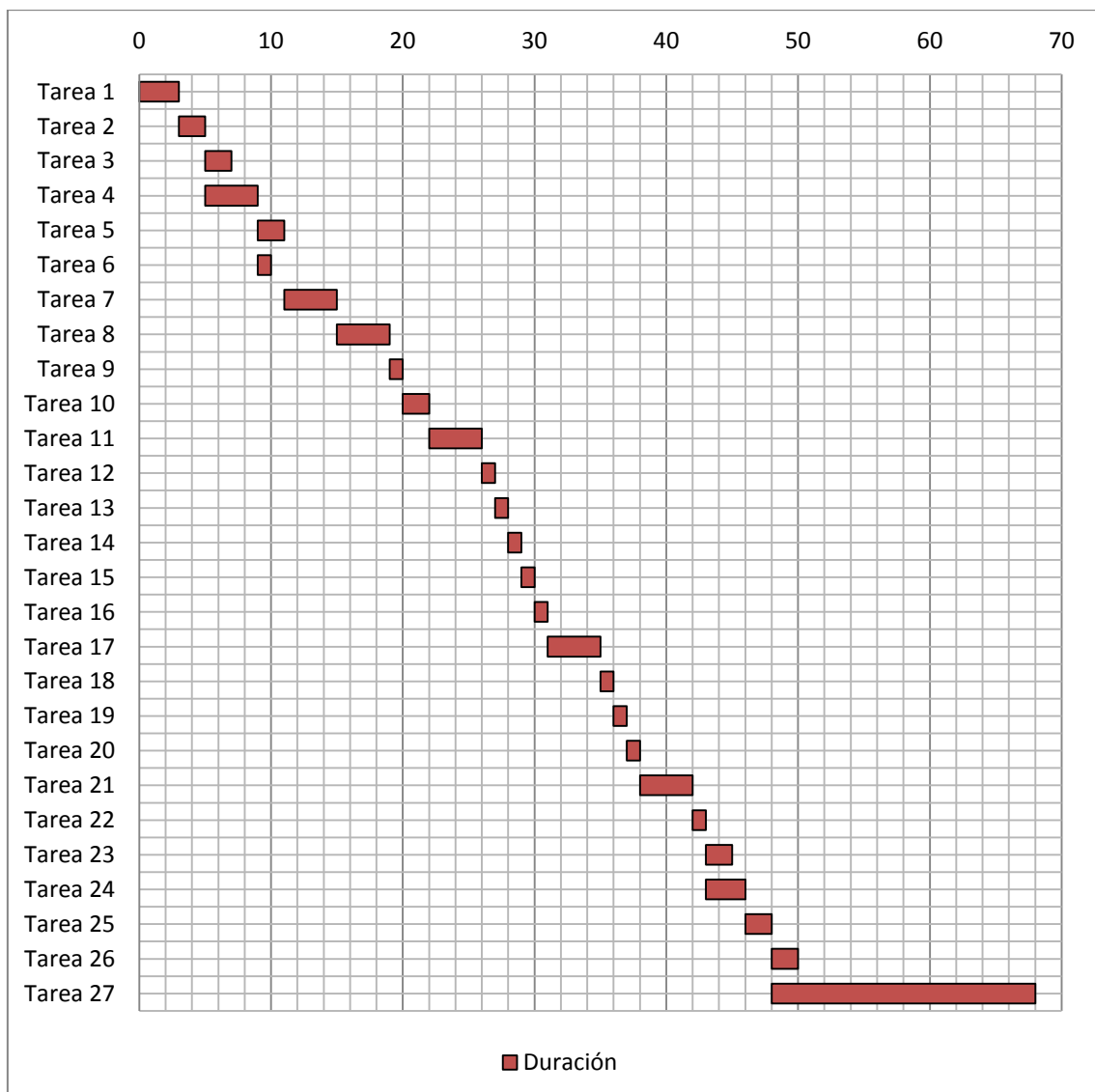
En el apartado anterior se explicaban las fases generales que han formado parte del proyecto. Estas fases se han dividido en sub-fases más pequeñas que se explican a continuación continuadas del tiempo en el que se han llevado a cabo.

- Introducción en protocolos de encaminamiento centralizados basados en SDN.
 - o Búsqueda de documentación sobre SDN. (3 días) - Tarea 1

- Estudio de ventajas teóricas que ofrece SDN. (2 días) - Tarea 2
- Estudio del protocolo Openflow y controlador Pox para construir una red basada en SDN.
 - Búsqueda de protocolo que implemente SDN. (2 días) - Tarea 3
 - Análisis de requisitos para configurar pruebas. (4 días) - Tarea 4
 - Búsqueda de controlador. (2 días) - Tarea 5
 - Búsqueda de software para la construcción de redes Openflow virtualizadas. (1 día) - Tarea 6
 - Primeras pruebas de Openflow + POX. (4 días) - Tarea 7
 - Estudio de Openflow en redes físicas. (4 días) - Tarea 8
- Construcción de caso de uso 1.
 - Análisis de objetivos y planificación. (1 día). - Tarea 9
 - Entorno físico.
 - Montaje y configuración de entorno físico. (2 días) - Tarea 10
 - Construcción de controlador. (4 días) - Tarea 11
 - Prueba del entorno y análisis de resultados. (1 día) - Tarea 12
 - Entorno virtual.
 - Montaje y configuración de entorno virtual. (1 día) - Tarea 13
 - Prueba del entorno y análisis de resultados. (1 día) - Tarea 14
- Construcción de caso de uso 2.
 - Análisis de objetivos y planificación. (1 día). - Tarea 15
 - Entorno virtual
 - Montaje y configuración de entorno virtual. (1 día) - Tarea 16
 - Mejora del controlador (4 días). - Tarea 17
 - Prueba del entorno y análisis de resultados. (1 día) - Tarea 18
- Construcción de caso de uso 3.
 - Análisis de objetivos y planificación. (1 día). - Tarea 19
 - Entorno virtual
 - Montaje y configuración de entorno virtual. (1 día) - Tarea 20
 - Mejora del controlador (4 días). - Tarea 21
 - Prueba del entorno y análisis de resultados. (1 día) - Tarea 22
- Recopilación de resultados y datos.
 - Comparación de resultados esperados con resultados obtenidos. (2 días) - Tarea 23
 - Documentación de las pruebas realizadas. (3 días) - Tarea 24

- Confección de memoria.
 - Diseño de la estructura. (2 días) - Tarea 25
 - Consulta de estadísticas para construir el marco actual en el que se encuentran los operadores. (2 días) - Tarea 26
 - Escritura de la memoria. (20 días) - Tarea 27

A partir de esta planificación de fases y subfases y teniendo en cuenta que algunas pueden ser realizadas de forma simultánea se obtiene el siguiente diagrama de Gantt.



A partir del gráfico de Gantt se concluye que el desarrollo de este proyecto ha llevado un tiempo de 68 días durante los que se han cumplido los objetivos propuestos al principio de este documento.

8. Conclusiones

Antes de terminar, es importante realizar un análisis general de todo lo aprendido durante el desarrollo de la investigación para extraer conclusiones.

8.1. El futuro de SDN

Después de analizar qué es SDN y que ventajas e inconvenientes ofrece, podemos hacernos una idea de cómo serán las redes del futuro.

En primer lugar, se observa una clara tendencia, tanto por parte de los fabricantes de equipos de interconexión, como por parte de los operadores, a migrar sus filosofías de red a un modelo más flexible y que se adapte mucho mejor a las necesidades y demandas de los usuarios.

Sin embargo, esto no asegura el éxito de SDN como el modelo dominante, ya que los fabricantes de switches están realizando un gran esfuerzo en proponer alternativas que ofrezcan ventajas semejantes a SDN pero que mantengan el control de la red en los nodos de interconexión.

Lo que sí parece estar claro es que los operadores necesitan evolucionar sus redes a un modelo diferente que soporte las exigencias del futuro. Esta evolución además debe ser rápida para conseguir modelos de negocio que aseguren la sostenibilidad de los negocios de los operadores. La tendencia de tráfico indica que en cuatro años el tráfico en Internet se triplicará y las redes deben estar preparadas para ello.

A través de los resultados de las pruebas realizadas durante el proyecto también nos damos cuenta de una cosa, cambiar de un modelo distribuido a un modelo centralizado y separar el plano de control del de datos aumentan significativamente la flexibilidad de la red permitiendo modificarla mucho más rápido.

Posiblemente esta (la flexibilidad) sea una de las condiciones principales que deberá cumplir la solución que se elija para las redes del futuro. En las redes actuales los grandes proveedores de datos pueden estar en cualquier punto (cada vez existen más servicios en la nube y más servicios multicast) y las redes deben actuar ante rápidos cambios de flujos.

SDN por su parte ya cumple esta condición de flexibilidad, por lo que es muy posible que se encuentre en la lucha por ser el modelo dominante en el futuro.

8.2. Aprendizaje adquirido

Sin duda, adquirir conocimientos en redes basadas en software hoy en día es importante en campos relacionados con redes informáticas. Aunque SDN no llegara a ser el modelo dominante, la solución seleccionada seguramente irá encaminada a pasar el control de la red al software.

Trabajar con redes semejantes a las del futuro permite estar preparado para nuevas exigencias por parte de las empresas y para tener una visión mucho más realista de cómo se trabajará en pocos años.

Además, en el desarrollo de la investigación se han adquirido competencias en planificación de proyectos, desarrollo en lenguaje Python, configuración de equipos de interconexión, gestión de tareas y capacidad de análisis.

Este aprendizaje adquirido de forma complementaria debe tenerse muy en cuenta ya que se trata de conocimientos generales que ayudan en el desarrollo de futuras investigaciones.

8.3. Próximos pasos

Como se ha detallado a lo largo de este trabajo, SDN es un paradigma que aún se encuentra en evolución y que todavía no está plenamente desarrollado. En este documento se ha realizado un examen general del estado actual de SDN y además se han estudiado las ventajas que ofrece OpenFlow (basado en SDN) de forma práctica, localizando algunas de las carencias que pueden ser objetivo de investigación.

Además de la mejora de las prestaciones que ofrece OpenFlow actualmente, otro punto en el que se podría continuar la investigación podría ser la mejora del controlador desarrollado. Obtener un controlador que cubra todas las necesidades actuales de una red real ayudaría a que SDN fuera implantado por empresas que necesitan una forma sencilla de gestionar su red. A continuación se exponen algunos puntos de investigación:

Desarrollo de API REST

Un punto muy importante de investigación sería el desarrollo de una API REST en Python que permita gestionar la red a través de software de terceros. Sería un gran avance ya que permitiría realizar cosas como:

- Crear webs desde las que gestionar los recursos de la red.
- Desarrollar servicios personalizados a los clientes en los que ellos decidan las características de red que quieren en cada momento.
- Crear algoritmos de encaminamiento sobre lenguajes diferentes a Python.
- Permitir a aplicaciones que necesitan un gran ancho de banda modificar la red cuando sea necesario.

Esta investigación sin duda haría que los desarrolladores de aplicaciones se fijaran en SDN y aceleraran la evolución del mismo.

Gestión de encendido/apagado de nodos

Hoy en día el ahorro de energía es un tema que se encuentra muy latente. La falta de recursos que producen energía y el empobrecimiento del medio ambiente son algunos de los motivos que han llevado a que cada vez sean más necesarias las soluciones que ahorren energía.

En el caso de SDN, un campo de investigación que ayudaría mucho a reducir la energía que consumen las redes sería el encendido y apagado de nodos en función de las necesidades de la red. Las redes de hoy en día realizan un gran consumo eléctrico debido a diferentes motivos:

- Las infraestructuras suelen estar replicadas para reducir la probabilidad de un corte del servicio. Esto consume bastante energía ya que hay que alimentar varias redes mientras solo se está usando una.
- Para refrigerar las salas en las que se alojan los nodos es necesaria refrigeración. Normalmente la mayoría de energía que se consume en una red se debe a la refrigeración de la misma.

SDN podría ser una solución a este problema ya que desde el controlador se tiene una visión global de la red. Llevando un control continuo del estado de la red en todo momento se podrían deshabilitar nodos e incluso zonas de la red rápidamente para ahorrar una gran cantidad del consumo.

La API de POX actualmente no tiene soporte para realizar este tipo de acciones, sin embargo, al ser un controlador de código abierto podría trabajarse en una nueva funcionalidad.

Creación de VPNs diferenciadas

El controlador desarrollado en este trabajo podría mejorarse implementando en el la posibilidad de definir distintas VPNs que convivan de forma conjunta en la red. Actualmente el controlador solo permite el uso de una.

Con esta mejora se podría dar la posibilidad de crear servicios diferentes que utilicen los recursos de la red de forma diferente. Así por ejemplo el servicio multimedia podría tener asignados los enlaces con menos jitter y menos retraso y los de servicios de mensajería y email los enlaces con mayor fiabilidad.

Esta mejora podría ser implementada con ayuda de la API actual de POX ya que permite establecer en los nodos de la red reglas diferentes para flujos con valor VPN diferente.

9. Bibliografía

- [1] Artículo: El tráfico de Internet se triplicará en cuatro años - http://www.tendencias21.net/El-trafico-de-Internet-se-triplicara-en-cuatro-anos_a34636.html
- [2] Requisitos videoconferencias con Skype a definición 1080p - http://logitech-es-emea.custhelp.com/app/answers/detail/a_id/43556/~requisitos-del-sistema-para-videoconferencias-1080p-con-skype-y-la-c%C3%A1mara-web
- [3] Tendencias en redes de altas prestaciones - <http://books.google.es/books?id=F9c1RheLgxC&lpg=PA115&ots=DUcCF5xKo4&dq=tráfico%20multimedia%20jitter&hl=es&pg=PA115#v=onepage&q&f=false>
- [4] Latest Internet traffic stats - http://www.telco2.net/blog/2009/10/perhaps_the_most_important_cha_1.html
- [5] Web ONO, servicio 200Mb - <http://www.ono.es/empresas/combinados/fijo-1linea-internet-200mb-tarifa-plana-1giga/>
- [6] Web Movistar, servicio Macrolan para empresas - <http://www.movistar.es/empresas/para-tu-oficina/redes/ficha/empresas-macrolan-vpn>
- [7] Inter-Datacenter WAN with centralized TE using SDN and OpenFlow - <https://www.opennetworking.org/images/stories/downloads/sdn-resources/customer-case-studies/cs-googlesdn.pdf>
- [8] Observatorio Nacional de las Telecomunicaciones - Indicadores de hogares con acceso a ancho de banda sobre hogares conectados a Internet - <http://www.ontsi.red.es/ontsi/es/indicador/hogares-con-acceso-de-banda-ancha-sobre-hogares-conectados-internet-incluyendo-adsl-red-de>
- [9] Observatorio Nacional de las Telecomunicaciones – Usuarios regulares de Internet - <http://www.ontsi.red.es/ontsi/es/indicador/individuos-que-usan-regularmente-internet>
- [10] Ericsson Mobility Report - <http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-november-2013.pdf>
- [11] Artículo: SDN, el futuro definido por software - <http://www.aunclidelastic.com/sdn-el-futuro-definido-por-software/>

- [12] Artículo: SDN, las redes al servicio de las necesidades del negocio -
<http://www.aunclidelastic.com/sdn-las-redes-al-servicio-de-las-necesidades-del-negocio/>
- [13] SDN at Google - Opportunities for WAN Optimization -
<https://www.ietf.org/proceedings/84/slides/slides-84-sdnrg-4.pdf>
- [14] Software-Defined Networking: The New Norm for Networks -
<https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- [15] Artículo: OpFlex protocol for ACI offered to IETF, OpenDaylight -
<http://www.networkworld.com/article/2175716/lan-wan/cisco-reveals-openflow-sdn-killer.html>
- [16] OpFlex: An Open Policy Protocol -
<http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.html>
- [17] Wikipedia: Software Defined Software - http://en.wikipedia.org/wiki/Software-defined_networking
- [18] Web OpenFlow - <http://archive.openflow.org/wp/learnmore/>
- [19] Web Pox Contoller - <http://www.noxrepo.org/pox/about-pox/>
- [20] Web Pantou OpenWRT firmware -
http://archive.openflow.org/wk/index.php/Pantou:_OpenFlow_1.0_for_OpenWRT
- [21] Web OpenWRT firmware - <https://openwrt.org/>

NOTA: Todas estas referencias han sido revisadas durante Agosto y Septiembre de 2014 obteniendo datos e información que han ayudado al desarrollo de esta memoria.

10. Anexos

Anexo 1:

Flow 1 (h1 -> h2)		
Switch	Interfaz de Entrada	Interfaz de Salida
S3	1	2
S4	1	2
Flujo Amarillo		

Anexo 2:

Flow 1 (h7 -> h8) – Prioridad 1		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	2
S2	4	5
S4	2	1
Flujo Amarillo		

Flow 3 (h7 -> h8) – Prioridad 2		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	3
S3	4	5
S4	3	1
Flujo Azul		

Anexo 3:

Flow 1 (h7 -> h8) – Prioridad 3		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	2
S4	1	2
S2	3	1
Flujo Amarillo		

Flow 2 (h7 -> h8) – Prioridad 2		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	3
S5	2	4
S2	4	1
Flujo Azul		

Flow 3 (h7 -> h8) – Prioridad 1		
Switch	Interfaz de Entrada	Interfaz de Salida

S1	1	3
S5	2	6
S6	2	3
S4	4	2
S2	3	1
Flujo Gris		

Flow 1 (h7 -> h9) – Prioridad 3		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	2
S4	1	3
S3	2	1
Flujo Marrón		

Flow 2 (h7 -> h9) – Prioridad 2		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	3
S5	2	5
S3	3	1
Flujo Naranja		

Flow 3 (h7 -> h9) – Prioridad 1		
Switch	Interfaz de Entrada	Interfaz de Salida
S1	1	3
S5	2	6
S6	2	3
S4	4	3
S3	2	1
Flujo Negro		

Flow 1 (h8 -> h9) – Prioridad 3		
Switch	Interfaz de Entrada	Interfaz de Salida
S2	1	4
S5	4	5
S3	3	1
Flujo Rojo		

Flow 2 (h8 -> h9) – Prioridad 2		
Switch	Interfaz de Entrada	Interfaz de Salida
S2	1	3
S4	2	3
S3	2	1
Flujo Verde		

Flow 3 (h8 -> h9) – Prioridad 1		
Switch	Interfaz de Entrada	Interfaz de Salida

S2	1	4
S5	4	6
S6	2	3
S4	4	3
S3	2	1
Flujo Violeta		

* Los flujos configurados han sido configurados tanto para la ida como para la vuelta de los paquetes. Al configurarse en POX reglas unidireccionales, es necesario crear dos flujos, uno de ida y otro de vuelta.



Universidad
Carlos III de Madrid
www.uc3m.es

TRABAJO FIN DE GRADO

**Estudio aplicabilidad SDN en red de agregación
EoMPLS**

Resumen en Inglés

Autor: Jorge Prudencio Fernández

Titulación: Grado en Ingeniería Telemática

Tutor: Jose Alberto Díaz García

Director: Carlos Jesús Bernardos Cano

Fecha: 24 de Septiembre de 2014

Summary

The users and enterprises have changed their form of work in Internet on the last years. We have changed from a centralized model, where the big providers of data were on localized places of the network, to a distributed model, where we can see data providers on all places of Internet.

This model change is forcing to the operators to change their solutions. They are searching solutions for support the trends of the future. We can see statistics where say that in four years the Internet traffic will be triple and we have to be ready to support the data.

The Internet operators are searching a flexible model that permit them ensure their business and support fast changes on the actual networks. They need to accommodate the needs of users.

Actually the technology available is being less effective with the pass of the time. The systems are too rigid and they don't have enough dynamism to support fast changes on the network.

Here is where SDN appears, a new paradigm that changes completely the form of understand the networks. Until now, the access networks worked with a distributed form. The infrastructure is formed of switches that communicate each other. This cloud of switches controls the Internet traffic.

Now, this model is ceasing to be feasible. Each switch has to be configured individually to support changes on the network.

SDN offer a centralized model where we have a separation between data and control panel. This model uses a controller that lives on a concrete point of the network. This philosophy solves the flexibility problem of distributed networks. With a centralized model we have a full view of the network and we can do changes at the same time from the controller.

This project has as objective revise all the theoretical advantages of SDN with practical cases. The cases will be based on real operator networks to work in a more real scenario.

Introduction

Nowadays anyone doubts that technology and Internet are the two elements that our life is based actually and it will be based on the future. Communication, information, entertainment, security, banking, orientation, buying, statics, multimedia,... all these topics and more have been translated to Internet and now we don't understand a world without it.

As we can see, Internet is vital, and it's very important to ensure a continuous and quality service to the clients. More and more business use Internet as key of their activities and a problem on the service can cause a lot of losses.

However, we are only on the principle of a future based completely on Internet. The need of be connected and informed all day is very important for people.

To support this big future demand of Internet the operators have started to work and research forms to ensure the network access and reply to client needs. Generate this reply is not easy. Operators have to use solutions that allow generate a profitable business model. It has to follow the price trends and the user habits.

So, is very important to start the investigation of those technologies and network philosophies that allow to be prepared for the Internet future. The operators are now very important enterprises because they are the responsible of ensure the access to Internet to millions of users to a very basic service.

In this document it will be researched SDN, the new alternative that have as objective increase the traditional distributed model to an centralized system where we obtain a global vision of the network.

Conclusions

The future of SDN

After analyze what is SDN and what advantages and disadvantages offer (from the practice point), we can have an idea of how will be the future networks.

First, we see a clear trend. The interconnection equips manufacturers and the Internet operators want to migrate their systems to a more flexible model. They need a system that adapts their networks to the future needs and demands of users.

However, this not ensures the success of SDN as the dominant model. The interconnection equips manufacturers are doing a big effort to promote alternatives that offer similar advantages as SDN but that maintain the control of networks on the interconnection nodes (switches).

What does seem to be clear is that the operators need to evolve their networks to a different model to support the future exigencies. This evolution has to be fast to get business models that ensure the sustainability of the operator enterprises. The trend of traffic says that in four years the traffic will be three times bigger and the networks have to be ready to support it.

If we see the results of the tests realized in this project we realize another thing. Change form a distributed model to a centralized model and separate the control and data plane extend significantly the flexibility of the network allowing modify it faster.

Flexibility can be one of the most important conditions that will have to accomplish the solution that dominate the future. On the actual networks the big data providers can be on

any point (we have a lot of services on the cloud and a lot of multicast services) and they have to reply fast to the flow changes.

SDN accomplish with this condition of flexibility, so is very possible that it fight for be the dominant model.